



## Stochastic optimal control of single neuron spike trains

Iolov, Alexandre; Ditlevsen, Susanne; Longtin, André

*Published in:*  
Journal of Neural Engineering

*DOI:*  
[10.1088/1741-2560/11/4/046004](https://doi.org/10.1088/1741-2560/11/4/046004)

*Publication date:*  
2014

*Document version*  
Publisher's PDF, also known as Version of record

*Citation for published version (APA):*  
Iolov, A., Ditlevsen, S., & Longtin, A. (2014). Stochastic optimal control of single neuron spike trains. *Journal of Neural Engineering*, 11(4), [046004]. <https://doi.org/10.1088/1741-2560/11/4/046004>

## Stochastic optimal control of single neuron spike trains

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2014 J. Neural Eng. 11 046004

(<http://iopscience.iop.org/1741-2552/11/4/046004>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 130.225.98.216

This content was downloaded on 25/08/2014 at 08:17

Please note that [terms and conditions apply](#).

# Stochastic optimal control of single neuron spike trains

Alexandre Iolov<sup>1,2</sup>, Susanne Ditlevsen<sup>2</sup> and André Longtin<sup>1,3</sup>

<sup>1</sup> Department of Mathematics and Statistics, University of Ottawa, Ottawa, ON K1N 6N5, Canada

<sup>2</sup> Department of Mathematical Sciences, University of Copenhagen, DK-1165 Copenhagen, Denmark

<sup>3</sup> Department of Physics, University of Ottawa, Ottawa, ON K1N 6N5, Canada

E-mail: [aiolo040@uottawa.ca](mailto:aiolo040@uottawa.ca)

Received 18 March 2014

Accepted for publication 2 April 2014

Published 3 June 2014

## Abstract

**Objective.** External control of spike times in single neurons can reveal important information about a neuron's sub-threshold dynamics that lead to spiking, and has the potential to improve brain-machine interfaces and neural prostheses. The goal of this paper is the design of optimal electrical stimulation of a neuron to achieve a target spike train under the physiological constraint to not damage tissue. **Approach.** We pose a stochastic optimal control problem to precisely specify the spike times in a leaky integrate-and-fire (LIF) model of a neuron with noise assumed to be of intrinsic or synaptic origin. In particular, we allow for the noise to be of arbitrary intensity. The optimal control problem is solved using dynamic programming when the controller has access to the voltage (closed-loop control), and using a maximum principle for the transition density when the controller only has access to the spike times (open-loop control). **Main results.** We have developed a stochastic optimal control algorithm to obtain precise spike times. It is applicable in both the supra-threshold and sub-threshold regimes, under open-loop and closed-loop conditions and with an arbitrary noise intensity; the accuracy of control degrades with increasing intensity of the noise. Simulations show that our algorithms produce the desired results for the LIF model, but also for the case where the neuron dynamics are given by more complex models than the LIF model. This is illustrated explicitly using the Morris-Lecar spiking neuron model, for which an LIF approximation is first obtained from a spike sequence using a previously published method. We further show that a related control strategy based on the assumption that there is no noise performs poorly in comparison to our noise-based strategies. The algorithms are numerically intensive and may require efficiency refinements to achieve real-time control; in particular, the open-loop context is more numerically demanding than the closed-loop one. **Significance.** Our main contribution is the online feedback control of a noisy neuron through modulation of the input, taking into account physiological constraints on the control. A precise and robust targeting of neural activity based on stochastic optimal control has great potential for regulating neural activity in e.g. prosthetic applications and to improve our understanding of the basic mechanisms by which neuronal firing patterns can be controlled *in vivo*.

**Keywords:** noisy, leaky integrate-and-fire model, spike-time controller, Hamilton-Jacobi-Bellman PDE, maximum principle for PDEs

(Some figures may appear in colour only in the online journal)

## 1. Introduction

Manipulation of individual neurons through electrical stimulation provides a mean of controlling their spiking activity. In applications such as brain-machine interfaces and



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

neuroprosthetics, a common goal is to record from a neuron and interpret the firing activity. Conversely, one may wish to stimulate a cell in order that it produce a desired firing rate, either fixed or varying in time. It is known that many cells fire sequences of spikes where the spike times—rather than just the rate—matter to post-synaptic neurons. In this paper, we explore the possibility of controlling a neuron in a way that it generates a sequence of spike times close to a desired sequence. We consider this problem in the framework of stochastic optimal control and give both a feedback solution (closed-loop), when the cell voltage is explicitly observable, as well as an open-loop solution when only the occurrence of spikes is observable. Importantly, we allow for an arbitrary noise intensity.

Both theoretically and in practice, related problems have been addressed in the literature. One objective has been to obtain either minimum or maximum interspike intervals lengths, when the input is constrained to be between some prespecified upper and lower bounds, see e.g. [19, 20] for a mathematical treatment, or [6, 25, 32] in a neuronal context. Another objective is to break a pathological synchronous firing pattern in clusters of neurons, highly relevant for neurological disorders such as epilepsy and Parkinson's disease, [23, 24], see also [13].

Our objective, namely targeting exact spike times in single neurons, has been considered mainly in the open-loop context, and in either absence of or for small noise. In [1] they use the spike response model, [15], to control output target spike trains and implement their scheme on pyramidal cells in mouse cortical slices. Their method is numerically efficient and allows for the simultaneous control of many neurons. However, it strongly relies on the assumption that the noise (the value of  $\beta$  in equation (1) below) is small. They only work with open-loop control. The difference between the objective in [1] and what we consider below is that they maximize the probability of spiking at some given time  $t^*$ , whereas we minimize the mean squared difference between the realized spike time,  $T_{sp}$  and the desired,  $t^*$ . Moehlis *et al* [21] work with a phase response model to obtain spikes at exact times, while keeping the root mean square of the input to a minimum. A similar approach is taken in [8]. They do not consider noise, though, and only work with open-loop control. In [26], their methods are implemented on brain slices of pyramidal neurons of a rat's hippocampus. While the phase response curve model is a parsimonious and effective way to describe a neuron's response to a stimulus, it is only valid in the supra-threshold regime, where the unstimulated neuron is periodically spiking. Feng and Tuckwell [12] investigate the control of the firing times of a leaky integrate-and-fire (LIF) neuron by varying the intensity of a noise process that drives the voltage (there is no other intrinsic noise in the model). To obtain a given spike time, they choose the objective of minimizing the variance of the membrane potential at the desired spike time, while forcing the mean of the membrane potential at this time point to equal the threshold. This provides exact solutions since it does not involve first-passage times, but has the drawback that there is a non-negligible probability that the obtained spike time will be far from the desired spike time.

Our objective of imposing a certain timing sequence for the spike train using an externally applied control is obtained in both the closed- and open-loop settings, and we specifically include the noise in the calculations of the controls. We restrict the controlled input to stay within pre-specified bounds, and also include a cost function to minimize intervention. Our main contributions are that we specifically allow for a non-negligible noise component in the neural activity when calculating the control, and consider both open- and closed-loop controls. The noise component is given by a Wiener process with a noise intensity larger than zero. In particular, we do not restrict our attention to the autonomously-spiking, supra-threshold regime.

The paper is structured as follows: first, we describe the neuron model and formalize the control objective. Then, we describe a feedback-based solution, which assumes that the controller has detailed access to the voltage trajectory. Then, we relax the observation assumption so that the controller only has access to the spike times. Finally, we compare the two methods through simulations against a simple-minded control technique which ignores the stochastic input to the neuron.

## 2. Problem formulation

A basic but useful model for the neural membrane potential evolution is the noisy LIF model:

$$\begin{aligned} dX(t) &= \left( I_{\text{ext}}(t) - \frac{X(t)}{\tau_c} \right) dt + \beta dW, \\ X(0) &= 0, \\ X(T_{sp}) = x_{th} &\Rightarrow \{X(T_{sp}^+) = 0. \end{aligned} \quad (1)$$

Here,  $X(t)$  represents the membrane electric potential at time  $t$ , which in absence of input decays to 0 with a time constant of  $\tau_c$ ,  $dW$  is a Brownian motion increment scaled by  $\beta$  and  $I_{\text{ext}}(t)$  is the deterministic external input to the cell. Having last spiked at time 0, the potential hits  $x_{th}$  at some random time  $T_{sp}$ , the potential resets to 0 and the process starts all over again. We write  $T_{sp}^+$  for the limit from the right at  $T_{sp}$ . Throughout the paper, the threshold is set to one,  $x_{th} = 1$ .

Suppose that we have some control over the external current such that it can be decomposed as

$$I_{\text{ext}}(t) = \mu + \alpha(t), \quad (2)$$

where  $\mu$  is an uncontrollable, but constant, part of the external current and  $\alpha(t)$  is controllable, i.e., it can be chosen to achieve some goal. A natural goal is to attempt to control the spike time,  $T_{sp}$ . That is, how do we choose  $\alpha(t)$  such that  $T_{sp} \approx t^*$ , where  $t^*$  is the desired spike time. A natural optimal control objective to achieve this is the least squares solution

$$\alpha(\cdot) = \arg \min_{\alpha(\cdot)} \{ \mathbb{E}[(T_{sp} - t^*)^2] \}, \quad (3)$$

where expectation is taken with respect to the distribution of the trajectories of  $X$ .

Often, the control has certain constraints. The most common are simple box constraints:

$$\alpha(t) \in [\alpha_{\min}, \alpha_{\max}] \quad \forall t. \quad (4)$$

In addition to equation (3), we will also add to our objective a running energy cost based on the control. This regularizes the problem eliminating the subtleties of singular-control situations and it serves to avoid excessive control as well as to avoid excessive charge building up on the cell, see [1].

So, we seek an optimal control,  $\alpha^*$ , that solves

$$J[\alpha(\cdot)] = \mathbb{E} \left[ \epsilon \int_0^{T_{sp}} \alpha^2(s) ds + (T_{sp} - t^*)^2 \right] \quad (5)$$

$$\alpha^*(\cdot) = \arg \min_{\alpha(\cdot)} J[\alpha(\cdot)],$$

where  $\epsilon$  measures how much weight we put on minimizing the energy cost. If  $\epsilon = 0$ , then we do not care at all about the expended energy cost.

It will often be the case that  $\alpha$  is either a function or a value of that function at a particular time. This function could be random, i.e., a stochastic process, which is naturally the case when it is a function of the random realizations of  $X$ . We will try to make clear below when we are referring to  $\alpha$  as a function and when we are merely referring to its particular value at some particular time. For example, in equation (5),  $\alpha(\cdot)$  refers to the function, while  $\alpha(s)$  refers to that function's value, possibly random, at time  $s$ .

We will consider two control contexts—*closed-loop* and *open-loop* control. In closed-loop control, the value of  $X$  at time  $t$  is observable and can be used in determining the control. In open-loop control, only the spike times are observable. In the closed-loop context, we write the control as

$$\alpha = \alpha(x, t)$$

to indicate its dependence on  $X(t)$  and to express that it will be updated based on the time-course of  $X$ . In the open-loop context, we write

$$\alpha = \alpha(t)$$

to indicate that  $\alpha(\cdot)$  is decided for all times at time 0. The techniques used to obtain the optimal controls in the two scenarios will be different. For the closed-loop scenario, we use dynamic programming, see [14], while for the open-loop scenario, we use a form of the maximum principle applied to the transition density of the controlled process, see [5]. The transition density is the probability density function of the process being at a state  $y$  at time  $t$ , given it was at some state  $x$  at some earlier time  $s$ .

Crucially, we assume that the model parameters,  $\mu$ ,  $\tau_c$ ,  $\beta$  in equations (1) and (2) are known. This is a strong assumption and will be discussed later.

### 2.1. Parameter regimes

Different parameter regimes can be envisioned given equation (1), depending on whether the noise intensity,  $\beta$ , is relatively high or low, and whether the external, uncontrollable bias current,  $\mu$ , induces spikes in the absence of noise or not. Spikes will occur in the absence of noise, if and only if  $\mu > 1/\tau_c$ , which is called the supra-threshold regime. When  $\mu \leq 1/\tau_c$ ,

**Table 1.** Regime labels and example values. Note that for the numerical experiments below, we use  $\tau_c = 0.5$ .

$\mu$	$\beta$	
	1.5	0.3
$1.5/\tau_c$	Supra-threshold-high-noise	Supra-threshold-low-noise
$0.1/\tau_c$	Sub-threshold-high-noise	Sub-threshold-low-noise

the regime is called the sub-threshold regime. In addition, we will investigate two values of  $\beta$ , which we will call high-noise and low-noise, respectively. Example values for each parameter regime are given in table 1, and we visualize a single path for each regime in figure 1.

### 3. Closed-loop solution—dynamic programming

We now detail the dynamic programming approach to obtaining optimal feedback control. In closed-loop, the controller can be continuously updated depending on the realization of the stochastic process,  $X$ .

Given a time  $t$  and a value  $X(t) = x$  of the voltage, let  $(T_{sp} - t)$  be the unknown and remaining time to spike. Note that  $T_{sp}$  is a random variable. Given arbitrary  $t, x$ , our remaining-cost objective,  $J[\alpha(\cdot); x, t]$ , will be

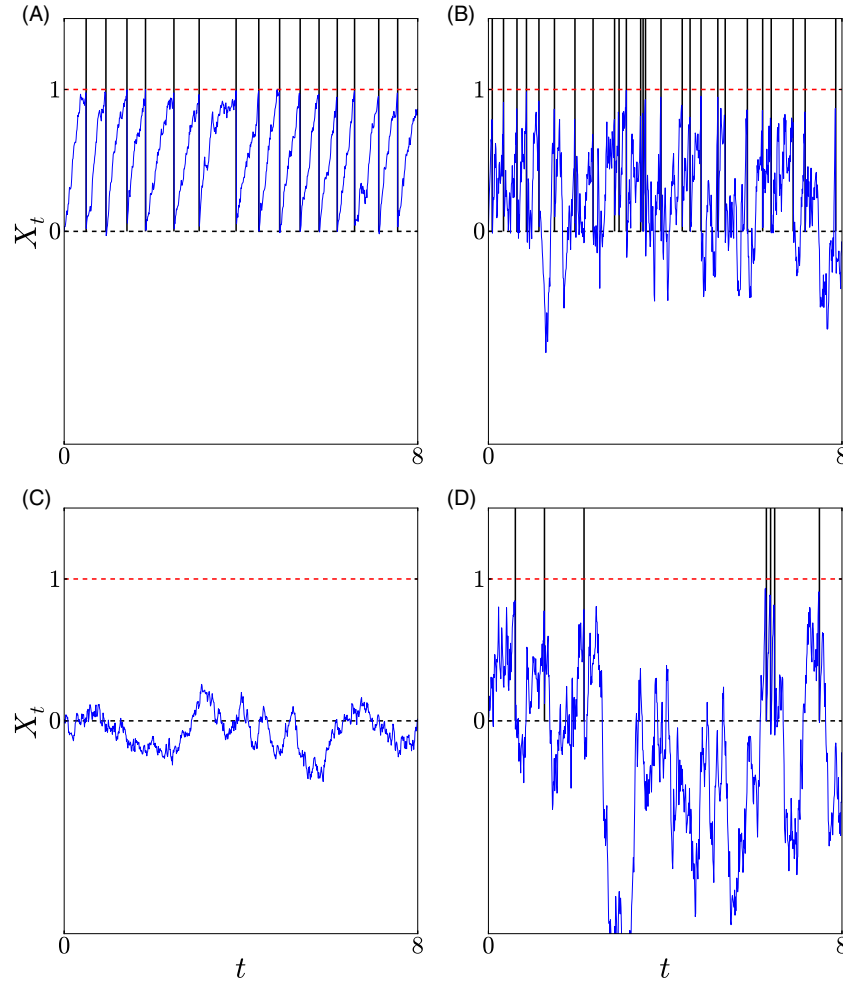
$$J[\alpha(\cdot); x, t] = \mathbb{E} \left[ ((T_{sp} - t) - (t^* - t))^2 + \epsilon \int_t^{T_{sp}} \alpha^2(s) ds \mid X(t) = x \right]. \quad (6)$$

That is, if time  $t$  has elapsed without a spike, we now want to minimize the difference between  $(T_{sp} - t)$  and  $(t^* - t)$ , given the current state  $x$ . The mean in equation (6) is taken over the distribution of hitting times,  $T_{sp}$ , conditional on  $X(t)$  or equivalently over the distribution of forward trajectories of  $X$  starting at  $x$  and ending at the threshold. Recall that in the closed-loop scenario, we assume that the value of  $X(t)$  is known to the controller. A similar problem is discussed analytically at length in the book on optimal control by Whittle [31], although there is no discussion there of the numerics required to solve it.

#### 3.1. Hamilton–Jacobi–Bellman equation

The Hamilton–Jacobi–Bellman (HJB) equation, see [14, 31] or the articles [6, 23] in a neuroscience context, associated with the optimal control for equation (6) is obtained as follows. We introduce the value function,  $w(x, t)$ , as the minimum of the remaining-cost objective, i.e., of the cost function between the current time  $t$  and the desired spike time  $t^*$ :

$$w(x, t) = \min_{\substack{\alpha(\cdot)_{s \geq t} \\ \alpha_{\min} \leq \alpha(\cdot) \leq \alpha_{\max}}} \{J[\alpha(\cdot); x, t]\} \\ = \min_{\substack{\alpha(\cdot)_{s \geq t} \\ \alpha_{\min} \leq \alpha(\cdot) \leq \alpha_{\max}}} \mathbb{E} \left[ ((T_{sp} - t) - (t^* - t))^2 + \epsilon \int_t^{T_{sp}} \alpha^2(s) ds \mid X(t) = x \right]. \quad (7)$$



**Figure 1.** Example trajectories from equation (1) using the parameter values from table 1. (A) Supra-threshold-low-noise. (B) Supra-threshold-high-noise. (C) Sub-threshold-low-noise. (D) Sub-threshold-high-noise. Note the multiple crossings very close together in the high-noise regimes in B and D.

Then,  $w$  satisfies the following HJB partial differential equation (PDE):

$$\partial_t w(x, t) + \frac{\beta^2}{2} \partial_x^2 w(x, t) + \min_{\alpha(x, t) \in [\alpha_{\min}, \alpha_{\max}]} \left\{ \epsilon \alpha^2(x, t) + \left( \mu + \alpha(x, t) - \frac{x}{\tau_c} \right) \partial_x w(x, t) \right\} = 0. \quad (8)$$

The special feature of equation (8) in contrast to a generic parabolic PDE is that it contains an embedded optimization that depends on the solution,  $w$ . For each  $x, t$  in the computational domain,  $\alpha$  is chosen such as to minimize  $\{\epsilon \alpha^2 + (\mu + \alpha - \frac{x}{\tau_c}) \partial_x w\}$ . Here, we can solve for the optimal control,  $\alpha^*(x, t)$ , analytically as

$$\alpha^*(x, t) = \underset{\alpha \in [\alpha_{\min}, \alpha_{\max}]}{\operatorname{argmin}} \left\{ \epsilon \alpha^2 + \left( \mu + \alpha - \frac{x}{\tau_c} \right) \partial_x w \right\} = \min \left( \alpha_{\max}, \max \left( \alpha_{\min}, -\frac{\partial_x w(x, t)}{2\epsilon} \right) \right). \quad (9)$$

We need to consider boundary conditions (BCs) for  $w$ . If  $X(t) = x_{th}$ , then we have a spike now and  $T_{sp} = t$ . Thus,

$$w(x_{th}, t) = (t - t^*)^2.$$

At the threshold, the value function equals the squared difference between the desired spike time and the realized one.

For large, negative values of  $x$ , we assume that  $w$  is not significantly affected by the change in  $x$ , i.e., that

$$\partial_x w(x_-, t) = 0$$

for some lower boundary  $x_-$ . Such a BC will be justified if we choose  $x_-$  such that the probability for the process to take values smaller than  $x_-$  is small. For example, we can take  $x_-$  to be two standard deviations below the mean of the stationary distribution of the maximally inhibited process, i.e., setting  $\alpha = \alpha_{\min}$  in equation (2). That is, we set  $x_- = \tau_c(\mu + \alpha_{\min}) - 2\beta/\sqrt{\tau_c/2}$ . We further enforce that  $x_- \leq -0.5$ .

Note that dynamic programming and the HJB equation work backwards. Thus, the evolution of the value function proceeds from the future to the past and we need some terminal condition (TC) at some point in the future, possibly infinity, from which to start incrementing  $w$  using the dynamics and the BCs. To determine TCs for  $w$ , our idea is simple: if we



reach  $t^*$  without having spiked we apply maximum control in the positive direction, i.e.,

$$t > t^* \Rightarrow \alpha(t) = \alpha_{\max}.$$

Thus,

$$w(x, t^*) = \mathbb{E}[(T_{\text{sp}} - t^*)^2 | X(t^*) = x, \alpha(t) = \alpha_{\max}]. \quad (10)$$

Note that we are making an approximation here—we are ignoring the energy term,  $\epsilon u^2$ , in the objective for  $t > t^*$ . Naturally, this approximation is ever more accurate for  $\epsilon \ll 1$ . We discuss in more detail the validity of the approximation in section 6. Alternatively, we could impose this approximate TC at some  $t^+ > t^*$ .

We will see the quantity on the right hand side of equation (10) repeatedly so we will give it a special name.

$$T_{(2)}(x) := \mathbb{E}[(T_{\text{sp}} - t^*)^2 | X(t^*) = x, \alpha(t) = \alpha_{\max}]. \quad (11)$$

$T_{(2)}$  is the second moment of the remaining time to reach the threshold starting at  $X(t^*) = x$  and applying  $\alpha_{\max}$  throughout. This quantity can be found easily for all  $x$  in the domain by solving a stationary backward Kolmogorov equation. This is an extension to the calculation of the first moment of an exit-time as seen in textbooks such as [17] and we give the details in appendix A.

Thus, we restate the HJB equation in its fully specified form:

$$\begin{aligned} \partial_t w(x, t) + \frac{\beta^2}{2} \partial_x^2 w(x, t) + \epsilon \alpha^2(x, t) \\ + \left( \mu + \alpha(x, t) - \frac{x}{\tau_c} \right) \partial_x w(x, t) = 0, \\ \alpha(x, t) = \min \left( \alpha_{\max}, \max \left( \alpha_{\min}, -\frac{\partial_x w(x, t)}{2\epsilon} \right) \right). \\ \begin{cases} w(x_{th}, t) = (t - t^*)^2 & \text{upper BC} \\ \partial_x w(x_-, t) = 0 & \text{lower BC} \\ w(x, t^*) = T_{(2)}(x) & \text{TC.} \end{cases} \end{aligned} \quad (12)$$

We are solving  $w(x, t)$  over the domain  $[x_-, x_{th}] \times [0, t^*]$ .

### 3.2. The numerical method for the HJB equation

We now have a PDE for  $w$  and an algorithm for computing all the BCs and TCs of this PDE. It is time to discuss the numerical method for solving equation (12). Since it is one-dimensional in space, it is straightforward to apply the standard centred finite difference using the Crank–Nicholson scheme to step in time, see chapter 19.2 in [28]. To resolve the nonlinearity,  $(\partial_x w)^2$ , in the PDE, we treat it as a mixed implicit–explicit term

$$\begin{aligned} (\partial_x w(x, t_k))^2 &= \partial_x w(x, t_k) \cdot \partial_x w(x, t_k) \\ &\approx \underbrace{\partial_x w(x, t_k)}_{\text{implicit}} \cdot \underbrace{\partial_x w(x, t_{k+1})}_{\text{explicit}}. \end{aligned}$$

Note that the implicit term is in the previous time  $t_k$  instead of, as is conventional, the next  $t_{k+1}$ , because we are solving for  $w$  backwards in time from  $t_{k+1}$  to  $t_k$  down to  $t_0 = 0$ .

The numerical scheme is implemented in Python, using the Scipy/Numpy library, [18]. For the discretization in time and space, we choose  $\Delta x$  and  $\Delta t$  in relation to the parameter

regimes. In particular, we take  $\Delta x$  to be less than  $\beta$  divided by the largest possible absolute value of  $(\mu + \alpha - x/\tau_c)$  for  $\alpha \in [\alpha_{\min}, \alpha_{\max}]$ ,  $x \in [x_-, x_{th}]$ ; in our examples, this always equals  $\mu + \alpha_{\max} - x_-/\tau_c$ . This is an attempt to ensure that numerically we are in the diffusion-dominated regime. We set  $\Delta t$  to be twice the value of  $\Delta x$  divided by the largest possible absolute value of  $(\mu + \alpha - x/\tau_c)$ . For example, in the supra-threshold low-noise regime, we have  $\Delta x = 0.0043$  and  $\Delta t = 0.0013$ . In all regimes, our discretization results in point-wise convergence of up to at least three significant digits for the value function. We consider this to be a sufficiently fine discretization.

We next demonstrate solutions for the value function and the associated control law for a parameter set from each regime in table 1.

### 3.3. Solutions of the HJB equation

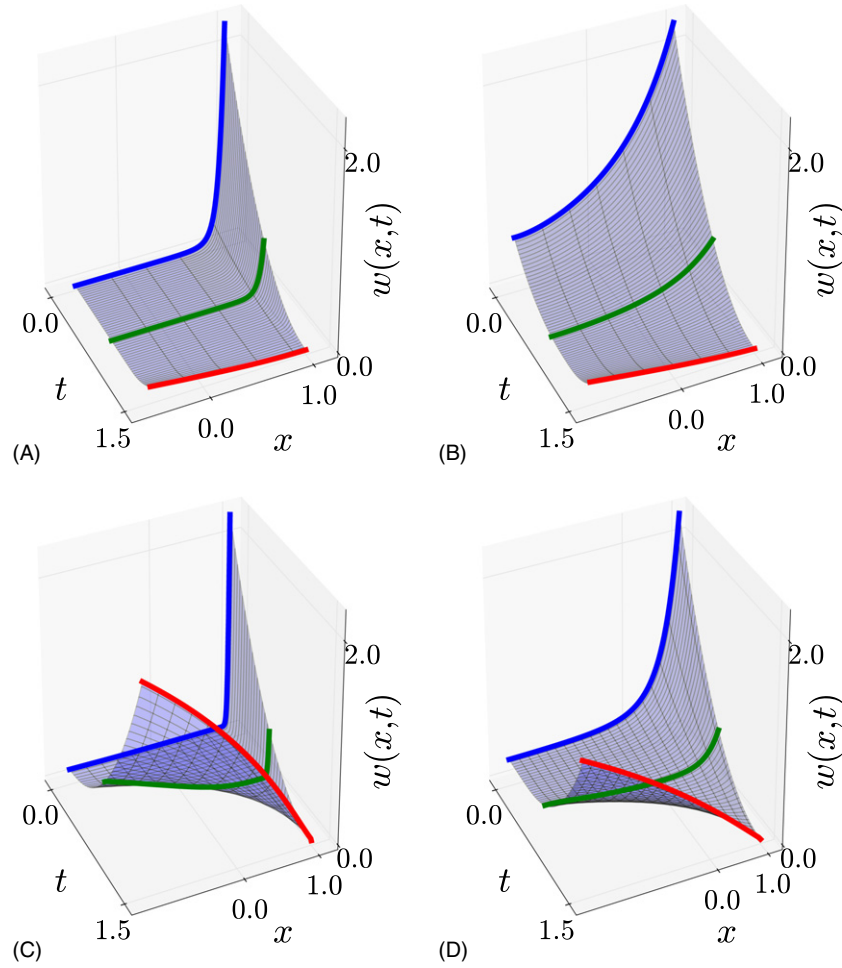
We set  $t^* = 1.5$  and  $\epsilon = 0.001$ , so that the primary focus is the accurate spiking and energy minimization considerations are secondary.

In figure 2, we show the computed value function,  $w(x, t)$ , the solution to equation (12), for each of the parameter regimes; while in figure 3, we show the corresponding surface plots of  $\alpha(x, t)$ .

Let us discuss the most salient features in figures 2 and 3. Recall that lower values for the value function,  $w(x, t)$ , are preferred, since we are minimizing. We see the same general shape in all four space-time surface plots of the value function in figure 2. At the end of the interval,  $w(x, t^*)$  is monotonically decreasing in  $x$ , which is to be expected given its TC in equation (10). That is, the lower the value of  $X(t^*)$  the longer on average will we have to wait for the spike to occur. As we go back in time,  $w$  inverts, with a clear peak near the upper threshold boundary. That is, for intermediate values of  $t$ , we have to consider the risk of spiking too early, represented by the high value of  $w$  near the upper boundary and the risk of spiking too late, represented by the high value of  $w$  at the lower boundary. As we progress even further back in time to the beginning of the interval, the peak near the threshold rises further, since we are spiking *earlier*, while the peak at the lower end flattens, since now there is enough time to reach threshold despite starting far from it. The full surface plots for  $w(x, t)$  in figure 2 can be thought of as the interpolation between these three basic phases.

Now focus on the controls in figure 3. Naturally, at the end of the interval, the control takes its maximum value,  $\alpha(x, t^*) = \alpha_{\max}$  for all  $x$ , i.e., it gives the maximum available push for the neuron to spike. This is consistent with equation (10). As we go back in time, however, the control  $\alpha$  decreases for  $x \ll x_{th}$ , and it becomes negative, i.e., inhibitory for  $x \lesssim x_{th}$ . That is intuitively consistent with the problem objective. For  $t < t^*$ , we want to bring  $X(t)$  closer to the threshold, but not too close, to avoid early spiking.

Now consider the differences between the individual regimes, i.e., the effect of changing the bias and the noise intensity,  $\mu$ ,  $\beta$ . All things being equal, the effect of increasing the noise intensity,  $\beta$ , is to lift the value function, i.e., to make



**Figure 2.** The numerical solution for the value function  $w(x, t)$  HJB PDE, equation (12), for the four different parameter regimes. The desired spike time is set to  $t^* = 1.5$ . The control bounds are  $\alpha \in [-2, 2]$ . The coloured contours for fixed  $t$  plotted as thick lines correspond to the beginning, middle and end times of the interval (blue:  $t = 0$ , green:  $t = t^*/2$ , red:  $t = t^*$ ). The energy penalty is  $\epsilon = 0.001$ . Note that all red curves for the value function go to zero at the threshold to satisfy the upper BC in equation (12). (A) Supra-threshold-low-noise. (B) Supra-threshold-high-noise. (C) Sub-threshold-low-noise. (D) Sub-threshold-high-noise.

our objective worse, at the beginning of the interval and to decrease it at the end. This is to be expected since we are attempting to minimize the variance in the spike time and without noise there would be no variance at all. However, at the end of the interval, noise only helps, since now we are only interested in spiking as soon as possible, and a higher noise intensity will tend to decrease the spike time on average. Increasing  $\beta$  also has the effect of increasing the size of the boundary layer near  $x_{th}$ , where the value function rises steeply. That is, for small  $\beta$ , that layer is small since the risk of spiking early is significant only close to the threshold. For larger  $\beta$ , this layer increases, see panels A, C, with a thin layer, versus B, D, with a thicker layer, in figure 2. Similarly, increasing the bias,  $\mu$ , tends to decrease the value function, especially at the end since that has the unequivocal effect of preventing late spikes.

#### 4. Open-loop stochastic control

When the value of  $X(t)$  is unobservable, the transition density can be used to perform the optimization. Since the transition

density follows a deterministic PDE, we apply a maximum principle for PDEs as a method of obtaining the optimal control, see [5] for details on optimization with PDEs, or the article [2], for optimization with a Fokker–Planck-type of system.

##### 4.1. Fokker–Planck equation for the state density evolution

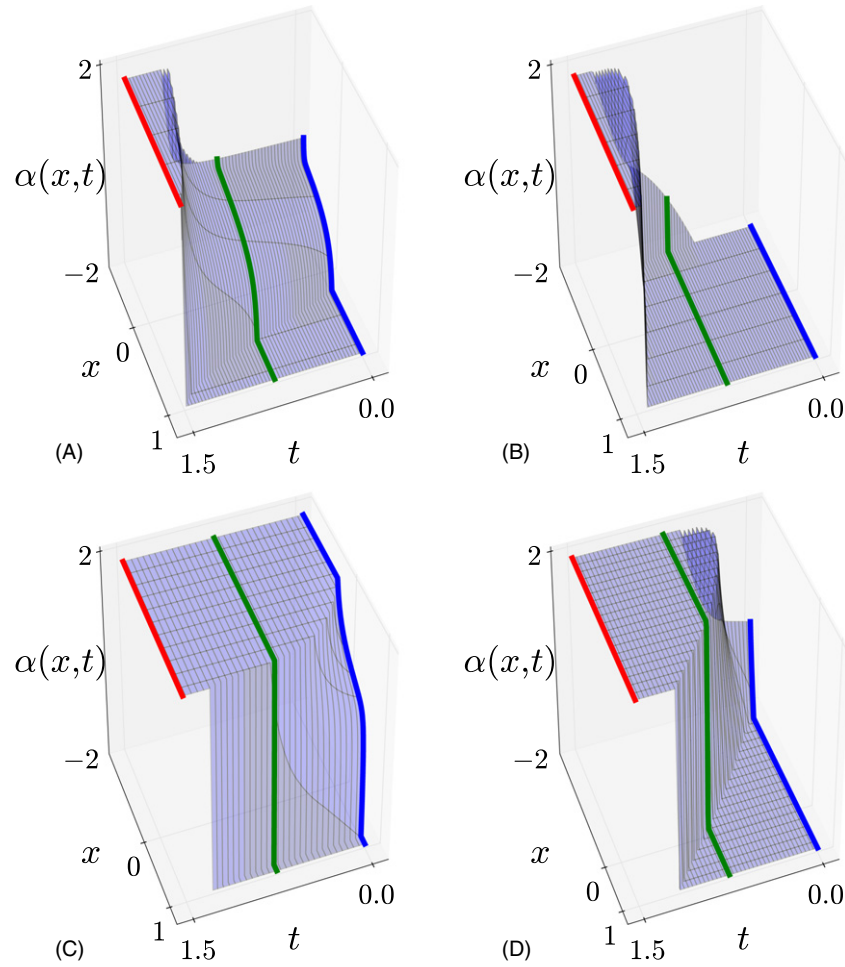
We write the transition density of  $X$ , conditional on no spikes having occurred since  $t = 0$ , as

$$f(x, t) dx = \mathbb{P}[X(t) \in dx | X(0) = 0, X(s) < x_{th} \quad \forall s < t].$$

Then,  $f$  satisfies a Fokker–Planck equation, see chapter 7 in [17],

$$\begin{aligned} \partial_t f(x, t) &= \frac{\beta^2}{2} \cdot \partial_x^2 f - \partial_x \left[ \left( \mu + \alpha(t) - \frac{x}{\tau_c} \right) \cdot f \right] \\ &\times \begin{cases} f(x, 0) = \delta(x) & \text{delta function} \\ \left( \mu + \alpha(t) - \frac{x}{\tau_c} \right) f - \partial_x \frac{\beta^2}{2} f \Big|_{x=x_-} \equiv 0 & \text{lower BCs at some } x_- \\ f(x, t)|_{x=x_{th}} \equiv 0 & \text{upper BCs at } x_{th}. \end{cases} \end{aligned} \quad (13)$$





**Figure 3.** Surface plots for  $\alpha(x, t)$  analogous to the surface plots for  $v(x, t)$  in figure 2. Note that the aspect of this figure is rotated relative to the surface plots of  $w(x, t)$  in figure 2, because the different surfaces are best seen from different angles.

In theory,  $x_- = -\infty$ , but in the numerics below, we will need to truncate it to some finite value, exactly as in the HJB equation, equation (12). Note that we write  $\alpha(t)$  here instead of  $\alpha(x, t)$ , since we cannot use the value of  $X(t)$ .

The  $f$  dynamics can also be written as

$$\partial_t f(x, t) = -\partial_x \phi(x, t)$$

for the probability flux

$$\phi(x, t) = \left( \mu + \alpha(t) - \frac{x}{\tau_c} \right) f - \partial_x \left[ \frac{\beta^2}{2} f \right].$$

Then, the lower BC is

$$\phi(x, t)|_{x=x_-} \equiv 0.$$

We will also need a short hand notation for the differential operator on the right side of equation (13). Let

$$\mathcal{L}_\alpha[\cdot] := \partial_x^2 \left[ \frac{\beta^2}{2} \cdot \right] - \partial_x \left[ \left( \mu + \alpha(t) - \frac{x}{\tau_c} \right) \cdot \right],$$

where  $[\cdot]$  indicates the argument of the operator  $\mathcal{L}_\alpha$ . We will usually omit the subscript  $\alpha$ , but have written it now to emphasize that the differential operator is parametrized by the control,  $\alpha$ .

#### 4.2. Restating the objective in terms of the transition density

We now derive the optimality conditions for the optimal control  $\alpha^*$  for the open-loop context. Recall our objective, equation (3). We can write this in terms of the transition density,  $f$ , as

$$\begin{aligned} J[\alpha(\cdot)] &= \int_{x_-}^{x_{th}} T_{(2)}(x) \cdot f(x, t^*) dx \\ &+ \int_0^{t^*} \phi(x_{th}, t) (t - t^*)^2 dt \\ &+ \epsilon \int_0^{T_{sp}} \alpha^2(t) \cdot \left( \int_{x_-}^{x_{th}} f(x, t) dx \right) dt. \end{aligned} \quad (14)$$

Let us explain in more detail each term on the right-hand side in equation (14).

The first term,

$$\int_{x_-}^{x_{th}} T_{(2)}(x) \cdot f(x, t^*) dx,$$

counts the cost of trajectories that spike too late. This cost is the expected squared time-to-hit starting at  $x$ , with  $\alpha = \alpha_{\max}$ , weighted by the probability of  $X(t^*) = x$ , which is just  $f(x, t^*)$ . Recall that  $T_{(2)}$  is defined in equation (11).

The second term,

$$\int_0^{t^*} \phi(x_{th}, t)(t - t^*)^2 dt,$$

counts the cost of trajectories that spike too early, that is the squared difference between some realized spike time,  $T_{sp} = t$ , and desired spike time  $t^*$ , weighted by the probability of a spike at  $t$ , which is just the outward probability flux at the threshold,  $\phi(x_{th}, t)$ . Recall that we assume that  $x_{th} = 1$  throughout. Note further that due to the homogeneous Dirichlet BC at  $x_{th}$ ,  $f(x_{th}) = 0$ , the outward flux is simply

$$\phi(x_{th}, t) = -\frac{\beta^2}{2} \partial_x f(x_{th}, t).$$

Finally, the third term,

$$\epsilon \int_0^{T_{sp}} \alpha^2(t) \cdot \left( \int_{x_-}^{x_{th}} f(x, t) dx \right) dt,$$

is the energy cost; the inner integral,  $\int_{x_-}^{x_{th}} f(x, t) dx$ , takes into account that we incur an energy cost only for those trajectories that have not yet spiked.

With that our optimal control  $\alpha^*(\cdot)$  will naturally be found via:

$$\alpha^*(\cdot) = \arg \min_{\alpha(\cdot)} J[\alpha(\cdot)].$$

#### 4.3. Optimizing using a maximum principle

By now, our problem of controlling the stochastic process in (1) has become a deterministic optimal control problem over its associated Fokker–Planck PDE. Our control  $\alpha$  influences the evolution density  $f$  and, via  $f$ , the integrals in the objective,  $J$ .

The maximum principle for PDEs, which is an extension to the famous Pontryagin maximum principle from finite dimensional systems, introduces an adjoint variable,  $p$ , which solves a PDE related to the PDE satisfied by the density  $f$  and then calculates the optimal control,  $\alpha(\cdot)$ , as a functional of  $f$  and  $p$ .

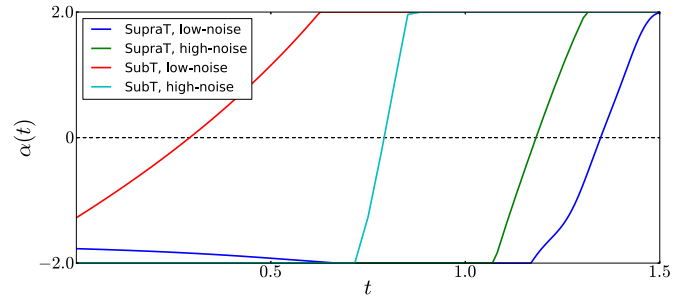
In short, the equation for the adjoint function,  $p$ , is

$$\begin{aligned} \partial_t p &= -\mathcal{L}^*[p] \\ &= -\left[ \frac{\beta^2}{2} \cdot \partial_x^2 p + \left( \mu + \alpha(t) - \frac{x}{\tau_c} \right) \cdot \partial_x p \right] \\ &\quad \times \begin{cases} p|_{x=x_{th}} = (t - t^*)^2 \\ \partial_x p|_{x=x_-} = 0 \\ p(x, t^*) = T_{(2)}(x) \end{cases} \end{aligned} \quad (15)$$

and then  $\alpha$  can be found via

$$\left\{ \epsilon 2\alpha(t) + pf \right\}_{x_-} - \int_{x_-}^{x_{th}} p \cdot \partial_x f dx = 0 \quad \forall t \in [0, T]. \quad (16)$$

Roughly speaking, equation (16) corresponds to setting the derivative of the objective  $J$  with respect to  $\alpha$  to zero, in equation (14) and the adjoint state,  $p$  acts as a Lagrange multiplier corresponding to the constraints of the transition density dynamics.



**Figure 4.** The deterministic optimal controls for each parameter regime as functions of time,  $t \in [0, t^*]$ . The desired spike time is set to  $t^* = 1.5$ , the energy penalty  $\epsilon = 0.001$  and the bounds are  $\alpha \in [-2, 2]$ . The initial value of the process is always the reset, i.e.,  $X_t = 0$  at time  $t = 0$ , see equation (1). From left-to-right, the control curves correspond to the sub-threshold-low-noise, sub-threshold-high-noise, supra-threshold-high-noise, supra-threshold-low-noise regimes.

More practically, the quantity

$$\begin{aligned} \delta_\alpha J(t) &= \left\{ - \int_{x_-}^{x_{th}} p(x, t) \cdot \partial_x f(x, t) dx \right. \\ &\quad \left. + p(x, t) f(x, t) \right\}_{x_-} + \epsilon 2\alpha(t) \end{aligned} \quad (17)$$

gives the direction of increase of  $J$  at  $\alpha(t)$  and we can use it as a gradient in a descent algorithm, given some initial guess,  $\alpha_0(t)$ , for the control.

Finally, we are ready to calculate the open-loop stochastic optimal control for the four parameter regimes. We use the same numerical method for solving the PDEs as described in section 3.2, since the PDEs for  $f, p$  are very similar in structure to the PDE for  $w$  except they do not contain a nonlinear term. The details of the simple gradient descent algorithm for obtaining the optimal open-loop control are given in algorithm B1. See [3] for a more sophisticated descent method based on the conjugate-gradient method. For our applications, the algorithm in B1 typically converges in less than 10 iterations, although sometimes it can take longer.

The optimal controls obtained using algorithm B1 for each regime are in figure 4. Recall that the optimal control,  $\alpha^*(t)$ , is open-loop, and thus, it is only a function of time.

Let us discuss the most salient features of the controls calculated in figure 4. In general, the strategy is to inhibit the neuron at the beginning of the interval,  $\alpha(t) = \alpha_{\min}$  for  $t \ll t^*$  and to excite it near the end,  $\alpha(t) = \alpha_{\max}$  for  $t \approx t^*$ . The smooth portion that connects these two segments in the middle of the interval is due to the energy penalty,  $\epsilon \int \alpha^2(s) ds$ . The behaviour of the control in different regimes is relatively simple to explain and we see what we would expect. In a sense, the bias current  $\mu$  can be absorbed by the control, and so, increasing  $\mu$  amounts to decreasing  $\alpha$  modulo its bounds. That is exactly the difference between the supra-threshold versus sub-threshold plots in figure 4 when holding the noise intensity fixed, either  $\beta = 0.3$  or  $\beta = 1.5$ . Both for low and high noise intensity, a reduction of  $\mu$  results in an increase in  $\alpha(t)$ . The effect of varying the noise intensity,  $\beta$ , is more subtle and it is not the same in the supra versus the sub-threshold regimes. In the supra-threshold regime, reducing  $\beta$  reduces the need for excitatory control towards the end, since the bias alone should

put the neuron over the threshold. Thus, the main part of the control with low noise in the supra-threshold regime is to stop the neuron from spiking too early. In the sub-threshold regime, the situation is somewhat reversed—reducing  $\beta$  obviates the need to apply an inhibitory control in the early part of the interval and also prompts the controller to apply excitatory input earlier, since it is the controller which is now the main drive for a spike.

We should mention that as  $\epsilon$  goes to zero, the optimal control should become bang–bang, meaning that at any time, it should assume either its minimum or maximum value. This is natural, since we have a classic situation where the dynamics are linear in the control and the cost is control independent (if  $\epsilon = 0$ ). This is also intuitive, since if there is no cost to the control, the controller would prevent early spiking by applying  $\alpha_{\min}$  and then close to the desired spike time, apply  $\alpha_{\max}$ . Theoretically, this could help in expediting the calculation of the open-loop control by reducing the problem to finding the optimal switch-point from  $\alpha_{\min}$  to  $\alpha_{\max}$ , but we have not pursued this further.

Finally, we would like to briefly compare conceptually the two optimal control approaches that we have used—dynamic programming and the maximum principle. The maximum principle essentially finds a critical point for the objective, akin to the first derivative test in calculus. Dynamic programming recursively builds up the optimal control from the end-time backwards, using a value function to represent the entire future cost associated with an incremental decision. The conceptual differences result in different numerical techniques. Most pertinently, when we apply dynamic programming, we can infer the optimal control right away, while for the maximum principle, we are forced to use an iterative, gradient-descent procedure.

## 5. Simulations

### 5.1. Single-spike control

Having obtained the optimal controllers, both closed- and open-loop, we evaluate their performance with simulated realizations of the voltage process, equation (1). In particular, while they both minimize the expected squared deviation of the spike time from some desired spike time, we analyse the distribution of the squared deviation and the behaviour of the controls for different parameter regimes.

In addition to the optimal controllers, we also show the behaviour of another control law—perhaps the most naive one—which is obtained by ignoring the noise and the energy penalty. That is, we find a *constant* value of  $\alpha$  that satisfies the desired BCs,  $x(0) = 0$ ,  $x(t^*) = x_{th}$ . This naive controller will be called ‘deterministic’, since it assumes deterministic dynamics in  $X$ , i.e.,  $\beta = 0$ .

For the comparison, we sample  $N$  realizations of the controlled system and apply in turn each of the three controls. Naturally, we reuse the same realization of the underlying stochastic process,  $W_t$ , for each of the three different controls.

We set  $\epsilon = 0.001$  and  $\alpha_{\max} = 2.0$ . As such, the energy cost is of secondary importance and the paramount effect on the objective is the squared difference,  $(T_{sp} - t^*)^2$ , which we are trying to minimize.

The performance of the three controllers for each parameter regime is given in table 2 and figure 5. Naturally, the closed-loop achieves a lower error than the open-loop, and the naive controller fares worst. The difference in performance mostly depends on the strength of the noise. Thus, for low noise, the performance of the stochastic controllers, be they open- or closed-loop, is not much superior to the naive deterministic controller. In contrast—in the high-noise regime, using a stochastic controller gives a much lower error on average between the desired  $t^*$  and the realized  $T_{sp}$ .

To give a better view of the performance of the controllers as a function of the noise intensity, we plot the ‘percentage of correct spikes’ and the ‘average squared spike-time deviation’ in figure 6. In figure 6, we also illustrate the effect of two values of the control bounds,  $\alpha_{\max} = 2$  or 4. Naturally, with a higher  $\alpha_{\max}$ , the error is reduced and the percentage of correct spikes is increased. As expected, the performance gets worse with increasing noise intensity. When the noise is small, the supra-threshold regime behaves best, but when the noise increases the sub-threshold regime provides more flexibility to correct for large perturbations caused by the noise.

For illustration sake, we also visualize some trajectories from the sub-threshold-high-noise regime in figure 7, see panels A, C, E. The most notable feature of the  $\alpha(\cdot)$  plots in figure 7 is that, especially for  $\epsilon = 0.001$ , there is a high level of fluctuations in the closed-loop  $\alpha$ . That is to be expected from the optimality condition in equation (9). Since the control is proportional to the derivative of the value function and we are minimizing, roughly speaking, the control attempts to push the process,  $X$ , to the bottom of the valley that is formed by the value function,  $w$ . However, the stochastic fluctuations of  $X(t)$  push it randomly to either side of that ‘valley’ and in turn force the control to change signs to counter-act. These fluctuations are then amplified by the division by the small parameter  $\epsilon$ , so that the control mostly bangs up or down to its extremes  $\alpha_{\min}$ ,  $\alpha_{\max}$ .

### 5.2. Multi-spike control

We now turn to the ultimate goal of our analysis, the control of spike *trains*. The major challenge when working with spike trains, that is not present when considering intervals in isolation, is that the target spike train might contain several closely clustered spike times, such that if the first controlled spike time occurs with delay, there will be too little time to hit the next target spike times and they will all be delayed. However, we adapt the controller to the actual controlled spike occurrences and compensate so that it is still targeting the originally posited train. Thus, there should be no accumulation of errors unless the controller is unable to catch up to an unusually high firing rate.

We proceed by generating  $M = 50$  realizations of  $N = 16$  spikes each in an attempt to meet a prescribed spike train. We focus on two regimes, the supra-threshold regime with either low or high noise. The results for parameters from the low-noise regime are shown in figures 8(a) and (b) and results for the high-noise regime are shown in figures 9(a) and (b). In all cases, the target train is obtained by a simulation of the

**Table 2.** Realized and theoretical performance of the different control laws. The empirical performance is obtained using  $N = 10\,000$  sample paths. The theoretical performance is found using the optimal value for  $J$  for the open-loop stochastic control and the value function  $w(x = 0, t = 0)$  for the closed-loop stochastic control.

Control law	Average squared spike-time deviation ( $N = 10\,000$ )	Expected squared spike-time deviation (theory)
Deterministic	0.287	–
Open-loop	0.003	0.008
Closed-loop	0.001	0.003
(a) Supra-threshold low-noise		
Control law	Average squared spike-time deviation ( $N = 10\,000$ )	Expected squared spike-time deviation (theory)
Deterministic	1.095	–
Open-loop	0.796	0.852
Closed-loop	0.795	0.843
(b) Supra-threshold high-noise		
Control law	Average squared spike-time deviation ( $N = 10\,000$ )	Expected squared spike-time deviation (theory)
Deterministic	0.327	–
Open-loop	0.142	0.150
Closed-loop	0.095	0.098
(c) Sub-threshold low-noise		
Control law	Average squared spike-time deviation ( $N = 10\,000$ )	Expected squared spike-time deviation (theory)
Deterministic	1.131	–
Open-loop	0.394	0.404
Closed-loop	0.360	0.365
(d) Sub-threshold high-noise		

supra-threshold high noise regime with no additional control, i.e., with  $\alpha(\cdot) = 0$ .

In each figure, figures 8(a), (b), 9(a) and (b), in the top panel, we show a smoothed version of the target train versus an empirical firing rate from the simulations. The red curve is a smoothed Gaussian kernel applied at the target times where the standard deviation of the kernel is  $\sigma = 0.1$ . The blue curve is the empirical firing rate equal to the number of spikes in a small window divided by the number of realizations and divided by the width of the window  $w_b$ . We have used  $w_b = 0.1$ . In the bottom panels of figures 8(a), (b), 9(a) and (b), we show in detail 10 out of the  $M = 50$  realizations.

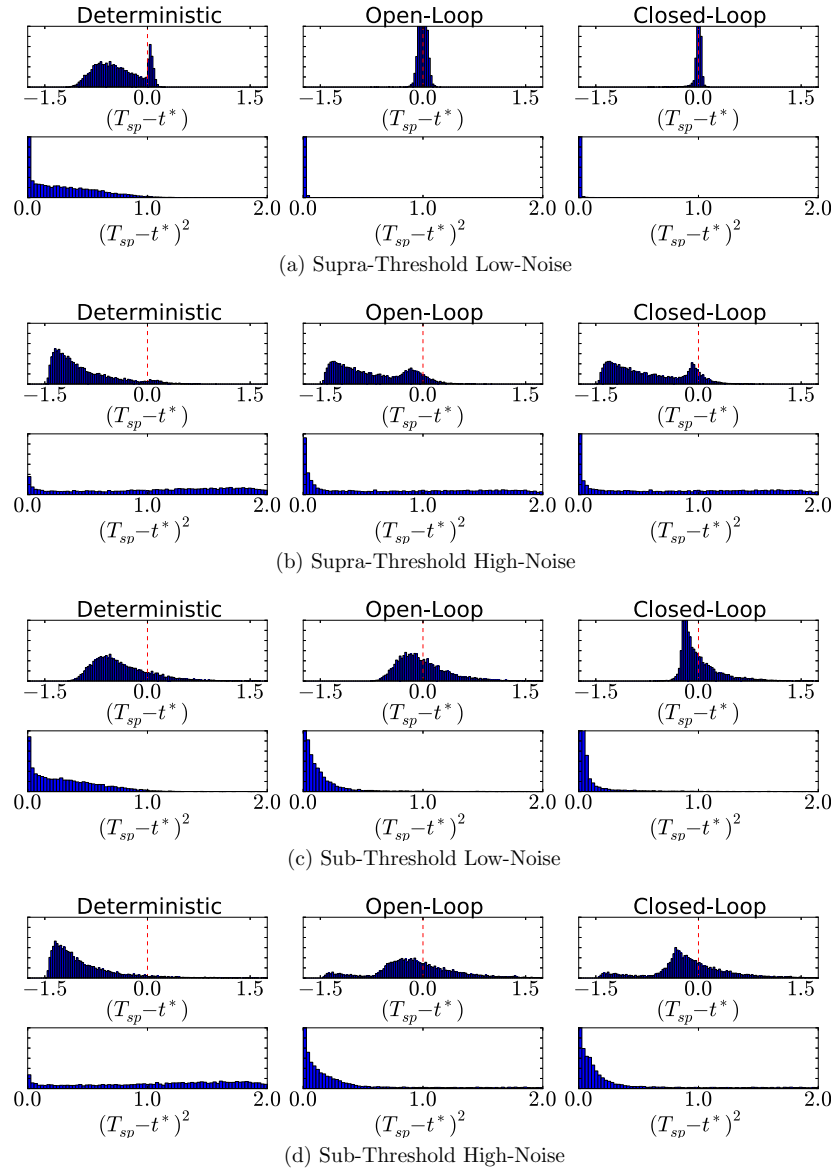
It is immediately clear that while the evoked trains closely follow the target train in the low-noise case, figures 8(a) and (b), this is a lot more difficult in the high-noise case, figures 9(a) and (b), and the restricted control is not able to produce reliable results. A stronger control is needed to obtain the same level of accuracy in the target. In figure 10, we relax the bound constraints on the control  $[\alpha_{\min}, \alpha_{\max}]$  from  $[-2; 2]$  to  $[-4; 4]$ . Then, significantly better tracking of the target train is achieved. Thus, to control a system under higher noise comes at the price of allowing a more powerful control signal.

The main question in this section is whether the separate optimization of each interval independently of the others is a suitable approach when targeting a whole spike train. In particular, we do not account for the possibility that two target spikes occur close together. Then, it might make sense to

attempt generating the first one a little earlier, on average, to give more time to generate the second. For example, in figures 8(a) and (b), the 13th target spike is successfully produced, but the 14th target spike occurs so soon after that in all trials it is delayed. However, if the target interspike intervals are within reasonable reach of the neuron, this is not a problem. Here, ‘reasonable reach’ is to be understood as any period longer than the time it takes to reach the threshold in the case of no noise and under maximum excitation, i.e.,  $\alpha(t) = \alpha_{\max}$ .

### 5.3. Controlling a biophysical model

To verify our method in a more biophysically realistic model, we use the established Morris–Lecar model [22], a Hodgkin–Huxley type of conductance-based model. To mimic an experimental situation, we will not assume that we know the model nor the parameters, and simply use the LIF model (1)–(2), where parameters are estimated from data sampled before the control is initiated. First, we estimate parameters for the LIF model from observations of the Morris–Lecar model. Then, we control the Morris–Lecar model using the scheme derived for the estimated LIF model. Thus, the control strategies will be conducted in a realistic experimental setting with no specific knowledge of the underlying mechanisms, considering data as if coming from a black-box.



**Figure 5.** Histogram of the spike timing error for the deterministic (left) versus open-loop stochastic (centre) versus closed-loop stochastic (right) control laws. Recall that  $T_{sp}$  is the random realized spike time and  $t^*$  is the target spike time. This is for the same problem as in figure 7. We have used  $N = 10000$  sample paths to form the statistics. (a) Supra-threshold low-noise. (b) Supra-threshold high-noise. (c) Sub-threshold low-noise. (d) Sub-threshold high-noise.

**5.3.1. The stochastic Morris–Lecar model.** The stochastic Morris–Lecar model including both current and channel noise is defined as the solution to

$$\begin{cases} dV_s = \frac{1}{C}(-g_{Ca}m_\infty(V_s)(V_s - V_{Ca}) - g_K W_s(V_s - V_K) \\ \quad - g_L(V_s - V_L) + I + A(s)) dt + \gamma d\tilde{B}_s, \\ dW_s = (\alpha(V_s)(1 - W_s) - \beta(V_s)W_s) dt + \sigma(V_s, W_s) dB_s, \end{cases} \quad (18)$$

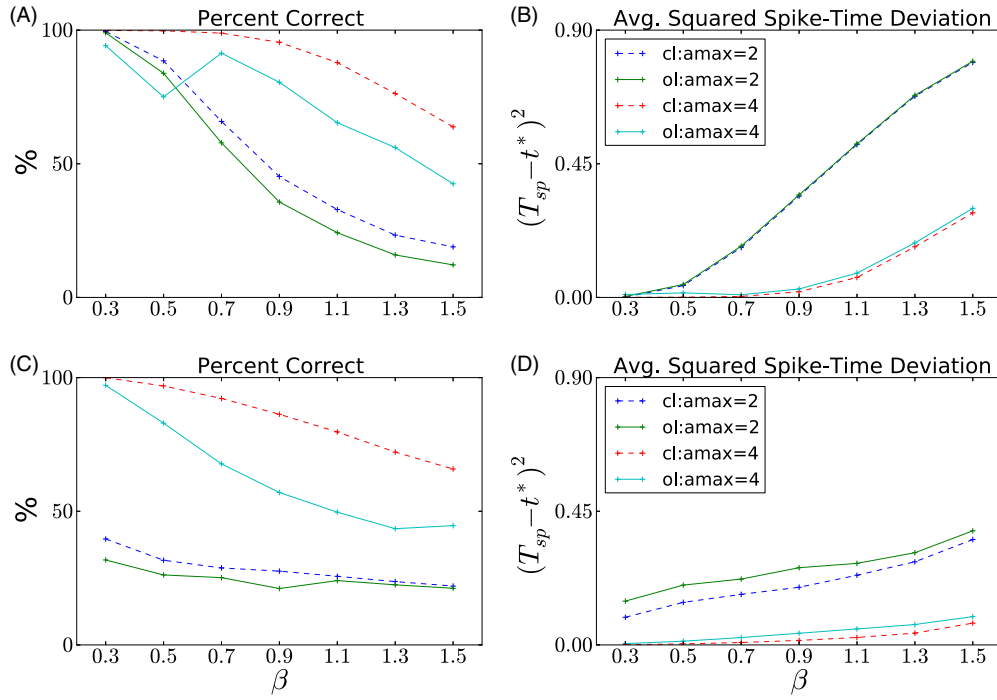
where

$$\begin{aligned} m_\infty(v) &= \frac{1}{2} \left( 1 + \tanh \left( \frac{v - V_1}{V_2} \right) \right), \\ \alpha(v) &= \frac{1}{2} \phi \cosh \left( \frac{v - V_3}{2V_4} \right) \left( 1 + \tanh \left( \frac{v - V_3}{V_4} \right) \right), \\ \beta(v) &= \frac{1}{2} \phi \cosh \left( \frac{v - V_3}{2V_4} \right) \left( 1 - \tanh \left( \frac{v - V_3}{V_4} \right) \right). \end{aligned}$$

The processes  $\tilde{B}_s$  and  $B_s$  are independent Brownian motions. The variable  $V_s$  represents the membrane potential of the neuron at time  $s$  and  $W_s$  represents the normalized conductance of the  $K^+$  current. It varies between 0 and 1, and can be interpreted as the probability that a  $K^+$  ion channel is open at time  $s$ . The equation for the dynamics of  $V_s$  contains four terms, corresponding to  $Ca^{2+}$  current,  $K^+$  current, a general leak current and the input current  $I$ . In addition, it contains the externally applied control current  $A(s)$ .

The functions  $\alpha(\cdot)$  and  $\beta(\cdot)$  model the rates of opening and closing of the  $K^+$  ion channels. The function  $m_\infty(\cdot)$  represents the equilibrium value of the normalized  $Ca^{2+}$  conductance for a given value of the membrane potential. The parameters  $V_1, V_2, V_3$  and  $V_4$  are scaling parameters;  $g_{Ca}, g_K$  and  $g_L$  are conductances associated with  $Ca^{2+}$ ,  $K^+$  and leak currents;  $V_{Ca}, V_K$  and  $V_L$  are reversal potentials for  $Ca^{2+}$ ,  $K^+$  and leak





**Figure 6.** The performance of the controllers (open-loop and closed-loop) as a function of the noise intensity  $\beta$ . The target spike time is again  $t^* = 1.5$ . On the left, panels (A), (C), we show the percentage of correct spikes, which is defined as the number of controlled spikes that occur within 10% of  $t^*$ , the target spike time. On the right, panels (B), (D), we show the average squared difference between the controlled spike and the target spike. Above, in (A), (B), we show the supra-threshold regime. Below, in (C), (D), we show the sub-threshold regime.

currents;  $C$  is the membrane capacitance; and  $\phi$  is a rate scaling parameter for the opening and closing of the  $K^+$  ion channels.

The parameter  $\gamma$  scales the additive current noise. Conductance fluctuations caused by random opening and closing of ion channels leads to multiplicative noise on the conductance equation. Function  $\sigma(V_s, W_s)$  models this channel or conductance noise. We consider the following function that ensures that  $W_s$  stays bounded in the unit interval if  $\sigma \leq 1$  [9]:

$$\sigma(V_s, W_s) = \sigma \sqrt{2 \frac{\alpha(V_s)\beta(V_s)}{\alpha(V_s) + \beta(V_s)} W_s(1 - W_s)}.$$

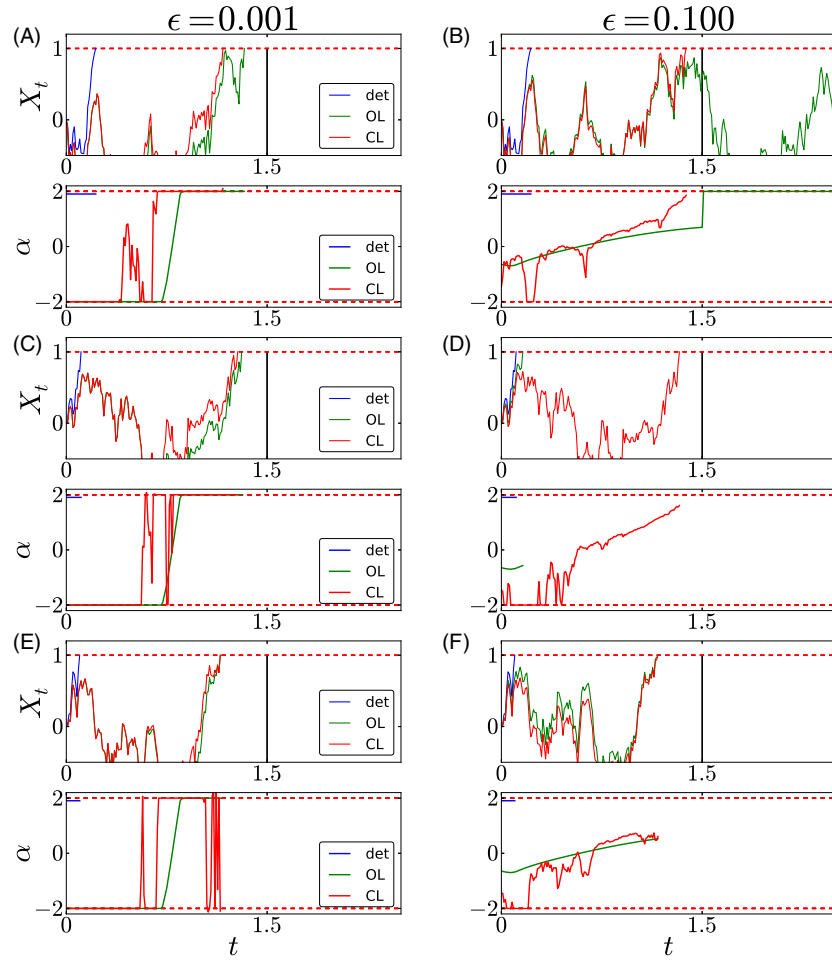
Parameter values used in the simulations are the same as those of [29, 30] for a *class I* membrane:  $V_K = -84$  mV,  $V_L = -60$  mV,  $V_{Ca} = 120$  mV,  $C = 20 \mu\text{F cm}^{-2}$ ,  $g_L = 2 \mu\text{S cm}^{-2}$ ,  $g_{Ca} = 4 \mu\text{S cm}^{-2}$ ,  $g_K = 8 \mu\text{S cm}^{-2}$ ,  $V_1 = -1.2$  mV,  $V_2 = 18$  mV,  $V_3 = 12$  mV,  $V_4 = 17.4$  mV,  $\phi = 0.07$ ,  $I = 40$  nA. The noise intensity values are taken from [11]:  $\gamma = 1$  mV s $^{-1/2}$ ,  $\sigma = 0.2$  mV s $^{-1/2}$ . Trajectories are simulated with an Euler–Maruyama scheme with a time step of 0.01 ms, which is then sub-sampled every ten points to compensate for approximation errors of the simulation scheme. Finally, we obtain a data set with observations every  $\Delta = 0.1$  ms. An example of a simulated trajectory is given in the top and bottom panels ( $V$ ,  $W$ , respectively) of figure 11. The peaks correspond to spikes of the neuron.

**5.3.2. Relating recorded voltage to a LIF model.** Assume discrete observations of the membrane potential, here

generated by the Morris–Lecar model as described above. These have to be related to the LIF model (1)–(2), which assumes that spikes are point events. Thus, the first problem is to partition the data into sub-threshold fluctuations and spikes. To make the method more robust, we operate with two thresholds. A lower threshold, where fluctuations below can be well approximated by the LIF model, is employed to identify the data used for estimation of LIF parameters. We set this to  $v_{th} = -22$  mV, see figure 11, upper and middle panels. A higher threshold is set to determine the start of the spike, a point of no return, from which the potential can only relax by going through a spike. We set this to  $-14$  mV, see figure 11, upper panel. When implementing the control assuming a LIF model, we attenuate the nonlinear effects of being close to spiking in the biophysical model (or real data) by treating the interval between  $-22$  and  $-14$  mV as a grey zone, artificially setting the data to a constant value just below the threshold. Finally, we need to fix the resetting of the voltage, which we set to  $v_r = -44.53$  mV.

In the Morris–Lecar, model the spike is not a point event, and in order to compare it to a LIF model with instantaneous reset, we ignore the voltage trajectory during spikes, where we will make no control and just wait for the reset. The effect of a spike for this set of parameters is around  $t_{ref} = 40$  ms, see lower panel of figure 11, at which point we restart. The time  $t_{ref}$  between the start of the spike and the reset is then entered in the LIF model as a refractory period. The value 40 ms is obtained from studying the voltage traces and is specific to the spike dynamics of the Morris–Lecar model. In a real experimental setting, it will probably be shorter, being the





**Figure 7.** Three examples for the controlled trajectories using the deterministic, open-loop stochastic and closed-loop stochastic control approaches. The plots on the left, (A), (C), (E), are obtained using a value of  $\epsilon = 0.001$ , while the plots on the right, (B), (D), (F), use  $\epsilon = 0.1$  (for a discussion on the effect of  $\epsilon$ , see section 6). The black vertical line in the plots indicates the desired spike time,  $t^*$ . The parameter values are  $\mu, \tau_c, \beta = [0.2, 0.5, 1.5]$  (sub-threshold high-noise regime), with  $t^* = 1.5$ . The upper plots show the voltage evolution,  $X(t)$ , while the lower plots show the applied control,  $\alpha(t)$ . Note that the optimal control,  $\alpha(t)$ , obtained from the deterministic and from the open-loop controls, is the same across all three samples.

sum of the spike duration and the refractory period, the exact duration will depend on the system.

The LIF model (1)–(2) is non-dimensional and normalized, resetting at 0 and spiking at a threshold of 1. Therefore, the data have to be transformed. First, we non-dimensionalize time by the transformation  $t = s/\bar{T}$ , where  $s$  is the time-scale of the measurements, and  $\bar{T}$  is some mean interspike interval; here, we use the average of the target spike trains,  $\bar{T} = 177.48$  ms. Then, we transform an observation  $V_s$  to  $X_t = (V_s - v_r)/(v_{th} - v_r)$ . On the transformed data, we estimate the parameters  $\mu, \tau_c$  and  $\beta$ . Then, we are ready to start the control. The control  $\alpha(t)$  is calculated on the transformed data, transformed back to a control on the original scale,  $A(s) = (v_{th} - v_r)\tau_c\alpha(t)$ , and fed into the Morris–Lecar model (or the cell in an experiment).

### 5.3.3. Estimating LIF parameters from Morris–Lecar data.

Assume  $\alpha(\cdot) = 0$  and discrete observations of model (1)–(2),  $X_n^{(k)}$ , for  $k = 1, \dots, K$ , where  $K$  is the number of interspike intervals, and  $n = 0, 1, \dots, N_k$ , where  $N_k + 1$  is

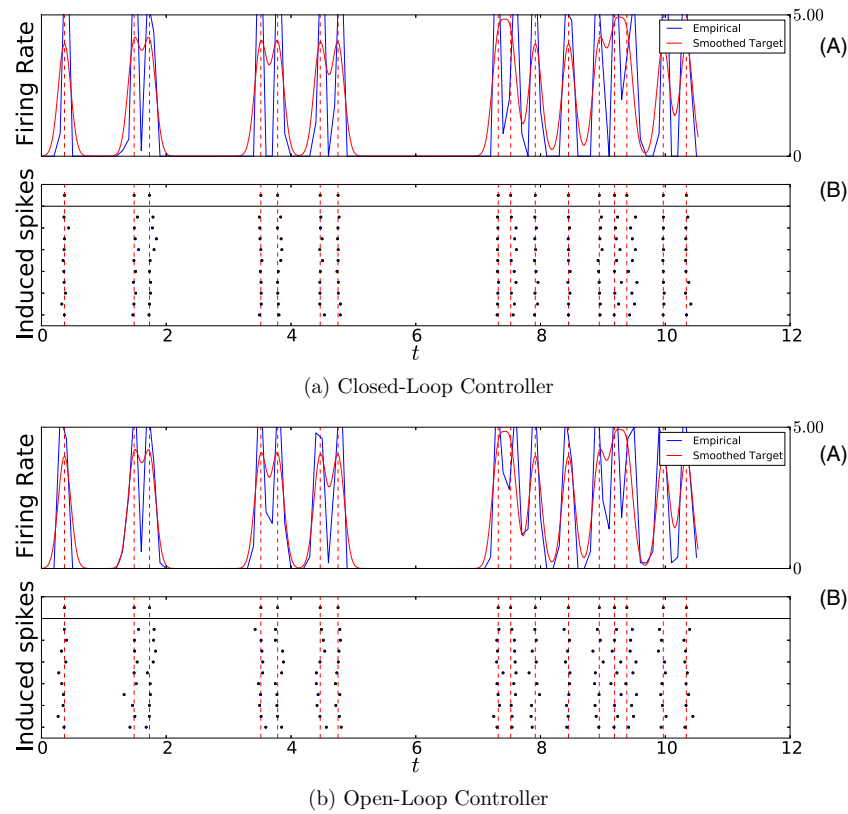
the number of observations in the  $k$ th interspike interval. The maximum likelihood estimators for the parameters  $\mu, \tau_c$  and  $\beta^2$  in absence of a threshold are given by, [4],

$$\begin{aligned}\hat{\mu} &= \frac{\sum_{k=1}^K \sum_{n=1}^{N_k} (X_n^{(k)} - e^{-\frac{\Delta}{\tau_c}} X_{n-1}^{(k)})}{N(1 - e^{-\frac{\Delta}{\tau_c}}) \hat{\tau}_c} \\ e^{-\frac{\Delta}{\tau_c}} &= \frac{\sum_{k=1}^K \sum_{n=1}^{N_k} (X_n^{(k)} - \hat{\tau}_c \hat{\mu})(X_{n-1}^{(k)} - \hat{\tau}_c \hat{\mu})}{\sum_{k=1}^K \sum_{n=1}^{N_k} (X_{n-1}^{(k)} - \hat{\tau}_c \hat{\mu})^2} \\ \hat{\beta}^2 &= \frac{2 \sum_{k=1}^K \sum_{n=1}^{N_k} (X_n^{(k)} - \hat{\mu} - (X_{n-1}^{(k)} - \hat{\mu})e^{-\frac{\Delta}{\tau_c}})^2}{N(1 - e^{-2\frac{\Delta}{\tau_c}}) \hat{\tau}_c}\end{aligned}$$

where  $N = \sum_{k=1}^K N_k$ .

Given Morris–Lecar data points over the time interval from 0 to 2 s, we get the following estimates:  $\hat{\mu} = 3.57$ ,  $\hat{\tau}_c = 0.20$  and  $\hat{\beta} = 0.59$ . For the bounds on our control, we set  $A(s) \in [\pm 10]$ , which corresponds to  $\alpha(t) \in [\pm 2.27]$ .

Now, we consider the more difficult case when only spike times are available, see [10] for a discussion. Estimating  $\tau_c$  from spikes is not accurate, it is (almost) unidentifiable.



**Figure 8.** Controller performance in the supra-threshold low-noise regime, given  $\alpha \in [-2, 2]$ . In (a) is depicted the closed-loop controller, in (b) the open-loop controller. In panels (A), the blue curve is the empirical firing rate of  $M = 50$  controlled trains, while the red curve is a smoothed version of the target train. (See text for details of how they are calculated.) In panels (B), the dashed, red lines indicate the target times,  $\tau_n$ , while the blue dots are the realizations of the controlled trains. The upper spike train above the solid line is the target spike train. The parameters from the supra-threshold low-noise regime are used, see table 1. The target spike train was generated using the model itself, with parameters from the supra-threshold high-noise regime, without an applied control,  $\alpha = 0$ .

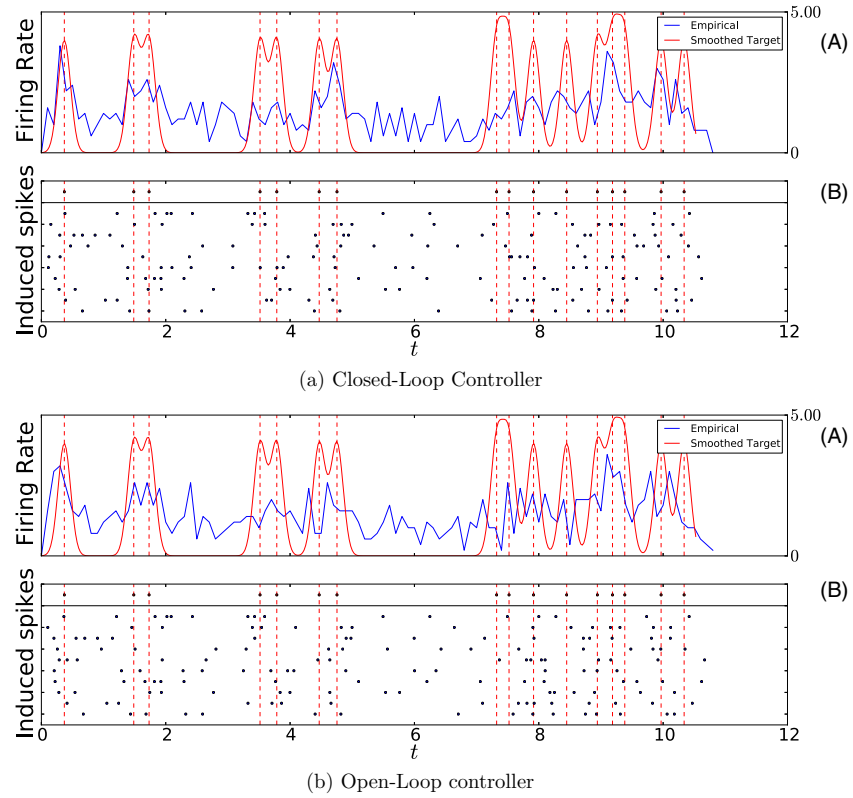
Therefore, we assume a fixed value of  $\tau_c = 0.11$ . This is quite different from the value of  $\tau_c$  estimated from intracellular recordings, so that we investigate robustness to misspecification of this value. Thus, we only estimate  $\mu$  and  $\beta$ , using the Fortet equation, for details see [10]. The estimates are  $\hat{\mu} = 6.17$  and  $\hat{\beta} = 0.78$ . This corresponds to  $\alpha(t) \in [\pm 3.94]$  if  $A(s) \in [\pm 10]$ .

**5.3.4. Controlling the Morris–Lecar model.** We now have all the pieces to control the Morris–Lecar model. In the closed-loop control strategy, we use the voltage-based estimates for the LIF model parameters, while for the open-loop strategy, we use the interval-based estimates. This is a disadvantage for the open-loop strategy, since on average, it will have poorer estimates of the real system, but provides a realistic picture of an experimental situation. The closed-loop results are shown in figure 12, and the open-loop results are shown in figure 13. In particular, the root mean-square error is 8.0 ms for the closed-loop controller and 23.2 ms for the open-loop controller, this corresponds to, respectively, 4.5% and 13.1% in relation to the mean target interval, which is 177.48 ms (This discards the  $t_{\text{ref}}$  portion of the interval).

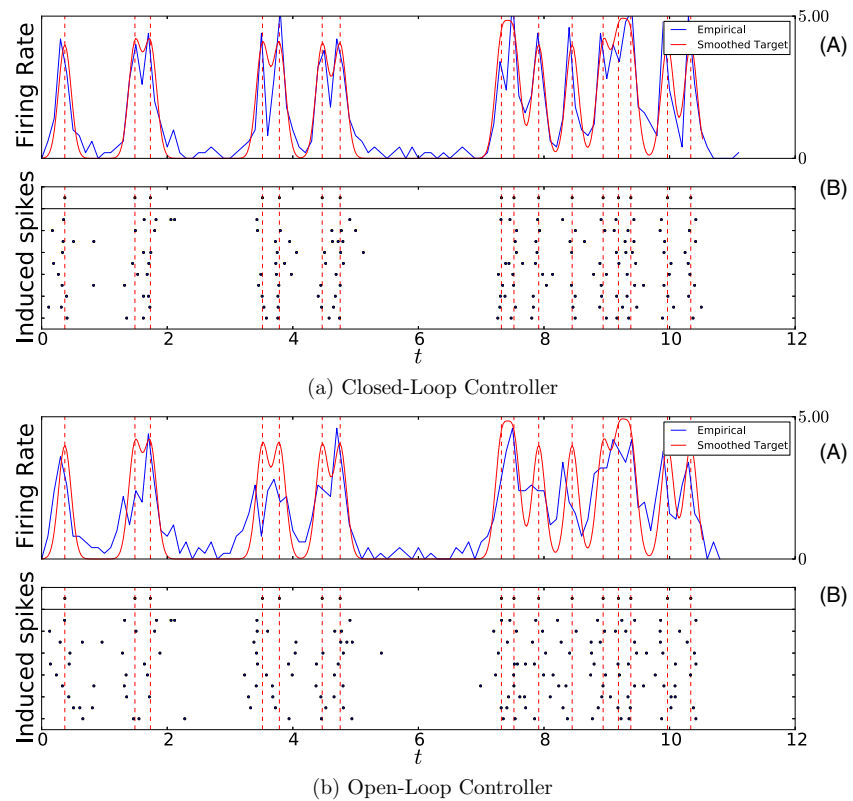
The results and accuracy are comparable to those in figure 8, and show that for the purpose of stochastic control,

it is not crucial to use a biophysical realistic model nor to know the parameters exactly. This is partly due to a lower value of  $\tau_c$  in the Morris–Lecar model, compared to what was used in the earlier simulations, and to slightly higher values of the bounds of the control. The variance is proportional to  $\tau_c \beta^2$ , which in the simulations is 1.125 (high noise) or 0.045 (low noise), and in the estimated Morris–Lecar is 0.07. There might also be some error cancelling, in the sense that the data provide estimated parameters that are the best choice for the misspecified model at hand, where errors are averaged out, thus providing more robust control.

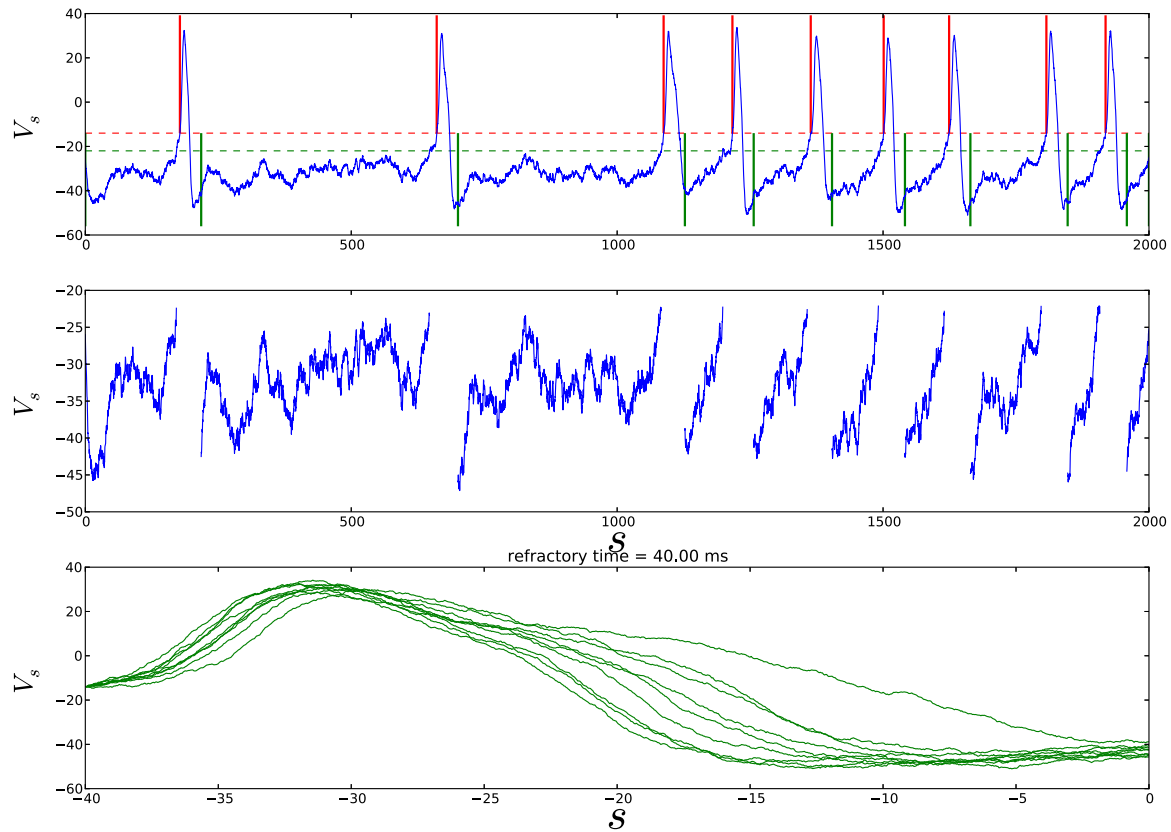
A similar result was obtained in [4], where the inhibitory and excitatory conductances were estimated from sub-threshold fluctuations using the basic LIF model, both on experimental and simulated data. Data were simulated from either a conductance-based model with Poisson synaptic input, or the more detailed two-compartment Booth–Rinzel–Kiehn model. In both cases, the LIF model, even if misspecified, resulted in sensible estimates of the conductances and their confidence intervals. It appears that the parsimony of the simple model provides robustness against nonlinear effects, space inhomogeneities and dynamics on various time scales.



**Figure 9.** Controller performance in the supra-threshold high-noise regime given  $\alpha \in [-2, 2]$ . In (a) is depicted the closed-loop controller, in (b) the open-loop controller. In panels (A), the blue curve is the empirical firing rate of  $M = 50$  controlled trains, while the red curve is a smoothed version of the target train. (See text for details of how they are calculated). In panels (B), the dashed, red lines indicate the target times,  $\tau_n$ , while the blue dots are the realizations of the controlled trains. The upper spike train above the solid line is the target spike train. The parameters from the supra-threshold low-noise regime are used, see table 1. The target spike train was generated using the model itself, with parameters from the supra-threshold high-noise regime, without an applied control,  $\alpha = 0$ .



**Figure 10.** Same as figure 9, but with the control constraints  $[\alpha_{\min}, \alpha_{\max}]$  increased to  $[-4, 4]$  instead of  $[-2, 2]$ .



**Figure 11.** Annotating (spike-segmenting) the raw Morris–Lecar data. The top panel shows a trajectory of  $V_s$  from the Morris–Lecar model. The middle plot shows the data segments that are used to estimate the LIF parameters from Morris–Lecar data. The bottom plot shows the stereotypical spike shape and justifies why 40 ms is a reasonable recovery time for this model with these parameters.

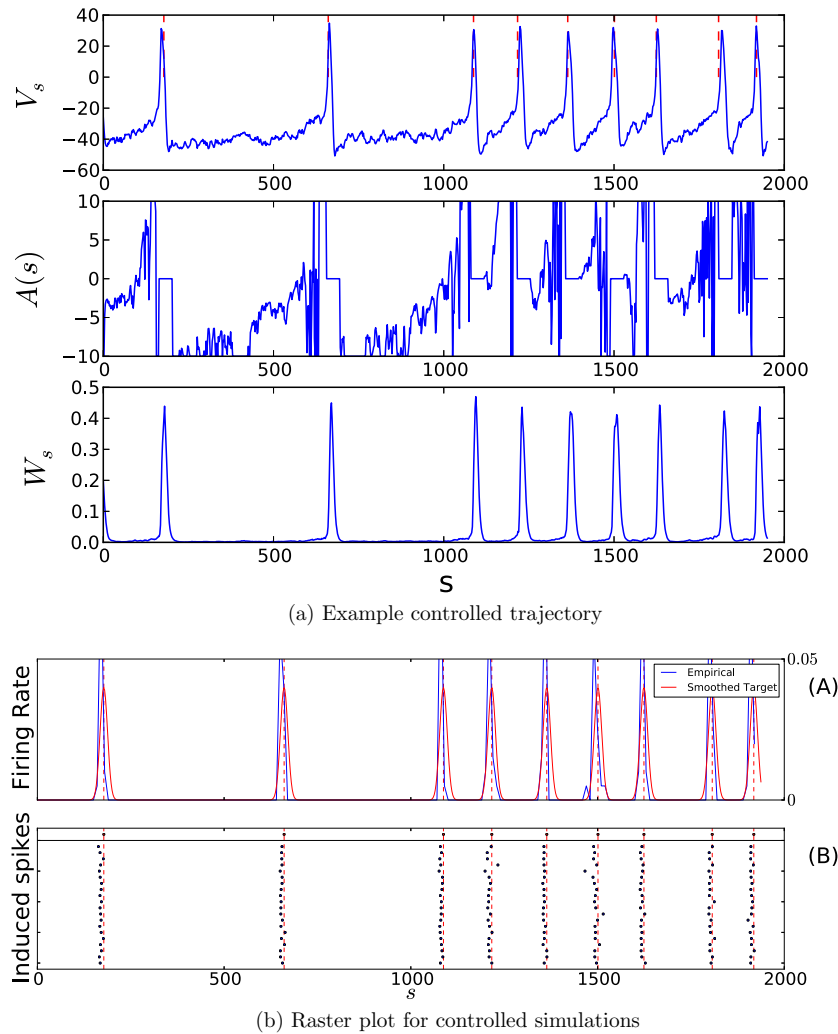
## 6. Effect of energy penalty

So far, we have assumed a low value for the energy penalty parameter,  $\epsilon$ , in the objective defined in equation (5). Our approach has been to be primarily concerned with the accurate spiking and that penalizing energy is rather intended to regularize the control and avoid excessive chattering of the control between its extreme values, than as a goal in minimizing energy in its own right. Now, we take some time to explore the effect of a higher  $\epsilon$  on the optimal controls. Intuitively, a higher value of  $\epsilon$  will tend to bring the optimal control,  $\alpha^*$ , closer to zero. We show the effect of setting  $\epsilon = 0.1$  in each of the four regimes for both the closed-loop control and the open-loop control in figures 14 and 15, respectively. These should be compared to figures 3 and 4, respectively. Also, in figure 7 on the right, we show example trajectories for the higher energy parameter value,  $\epsilon = 0.1$ , in the sub-threshold high-noise regime.

For both the closed-loop and open-loop optimal controls, increasing  $\epsilon$  tends to reduce the absolute value of  $\alpha^*(t)$ . In particular, for the closed-loop control, it tends to broaden the area of transition for decreasing  $x$  when the control swings from its minimal, i.e., inhibitory value, to its maximal, i.e., excitatory value. Similarly for the open-loop control, instead of banging from its minimal bound at the beginning of the interval to its maximal bound at the end, the optimal control

for  $\epsilon = 0.1$  tends to have a milder transition from a slightly inhibitory to slightly excitatory values.

Note however that increasing  $\epsilon$  has an important effect on the validity of the approximation used to form the TCs for the value function,  $w$ , and the adjoint variable,  $p$ . We assumed that applying  $\alpha_{\max}$  is optimal for  $t > t^*$ . This is indeed true in the limit  $\epsilon \rightarrow 0$ . However, even for a finite value of  $\epsilon$ , this may still be the optimal thing to do. Indeed, in the original calculations, where  $\epsilon = 0.001$ , see the panels on the left in figures 14 and 15, the so-obtained value function implied that  $\alpha(x, t^*) = \alpha_{\max}$ . In other words, our guess that  $\alpha(x, t) = \alpha_{\max}$  for  $t > t^*$ , is self-consistent with the so-obtained value function. This is akin to confirming an ansatz. For a higher value of  $\epsilon$ , like  $\epsilon = 0.1$ , this is no longer the case. The red curves on the right of figure 14 are no longer pushed up at  $\alpha = \alpha_{\max}$ , and as such it is no longer valid, strictly speaking, to assume that the value function can be obtained by assuming  $\alpha(t)|_{t>t^*} = \alpha_{\max}$  and then calculating the expected remaining time-to-spike. The exact same conclusion can be drawn from the open-loop control. There, the optimal control always (almost) reaches its maximum,  $\alpha_{\max}$ , while choosing  $\epsilon = 0.001$ , this is no longer so for the higher  $\epsilon = 0.1$ . In fact, as we mentioned after equation (10), the correct thing to do for higher values of  $\epsilon$  is to push the calculation interval to some  $t^+ > t^*$ , apply the same TC at  $t^+$  instead of  $t^*$  and then solve backwards (either for the closed- or open-loop control). One will be guaranteed



**Figure 12.** Closed-loop control of the Morris–Lecar dynamics. In (a) are plots of the trajectory and the control, in (b) is a raster plot of realized spikes. The upper spike train above the solid line is the target spike train.

to find some  $t^+ > t^*$  that works since eventually the quadratic term  $(t - t^*)^2$  will dominate the energy term in the objective, which is linear in  $t$ . We do not explore this further here.

## 7. Discussion

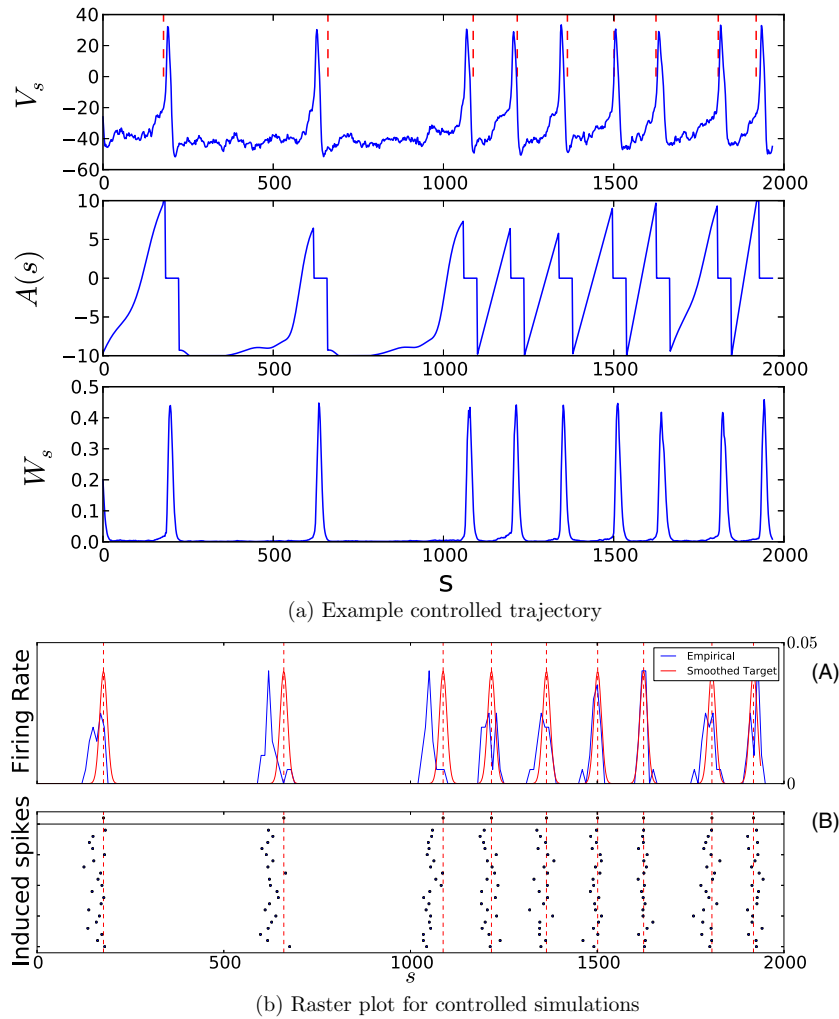
We have analysed and computed the optimal control of spikes in two scenarios—one where the underlying voltage of the neuron is observable to the controller (closed-loop control) and one where only the spikes are observable (open-loop control). The optimal control problem is solved using dynamic programming if the controller has access to the voltage (closed-loop control) and using a maximum principle for the transition density if the controller only has access to the spike times (open-loop control). We have further shown how one can implement both techniques for an arbitrary spiking neuron.

Our calculations in section 5.3 show that our choice of a simple model, the LIF, is quite robust, it can be correctly and quickly estimated from data and it allows for accurate control of realistic neurons with a significant noise intensity. This, together with our demonstration in section 5.1 that the

LIF control can accommodate a significant amount of noise, demonstrate the generality of our open-loop and closed-loop control methods to produce desired spike trains.

Naturally, the ability to control the system is most clearly affected by the level of the noise. In a sense, the noise acts like an adversary—in our context, it has no beneficial role, except to obstruct precise spiking. When the level of the noise is high, stronger control is needed, and the trade-off between accuracy (hitting the target spikes), possible damage to the system (the limits of the control) and energy expenditure has to be considered.

The bias current has a more helpful role, at least in our examples, where a high value of the bias tends to help precise spiking. Of course, where it helps or hinders depends on the value of the desired spike time—a combination of a high positive bias and a ‘distant’ spike time will tend to be difficult to control, as the system will naturally tend to spike earlier than desired. In summary, we have found that systems in the supra-threshold low-noise regime tend to be most accurately controlled, while the supra-threshold high-noise regime are the least accurate.



**Figure 13.** Open-loop control of the Morris–Lecar dynamics. In (a) are plots of the trajectory and the control, in (b) is a raster plot of realized spikes. The upper spike train above the solid line is the target spike train.

It should be noted that we do not require that the controlled cell is already in the supra-threshold regime—the control scheme applies in the same manner in the sub-threshold regime as well. It is only required that the maximum value of the control puts the model in the supra-threshold regime, implying that the control can always achieve supra-threshold behaviour. The supra- and sub-threshold distinction only applies to the intrinsic dynamics in the absence of control, and ignoring the noise.

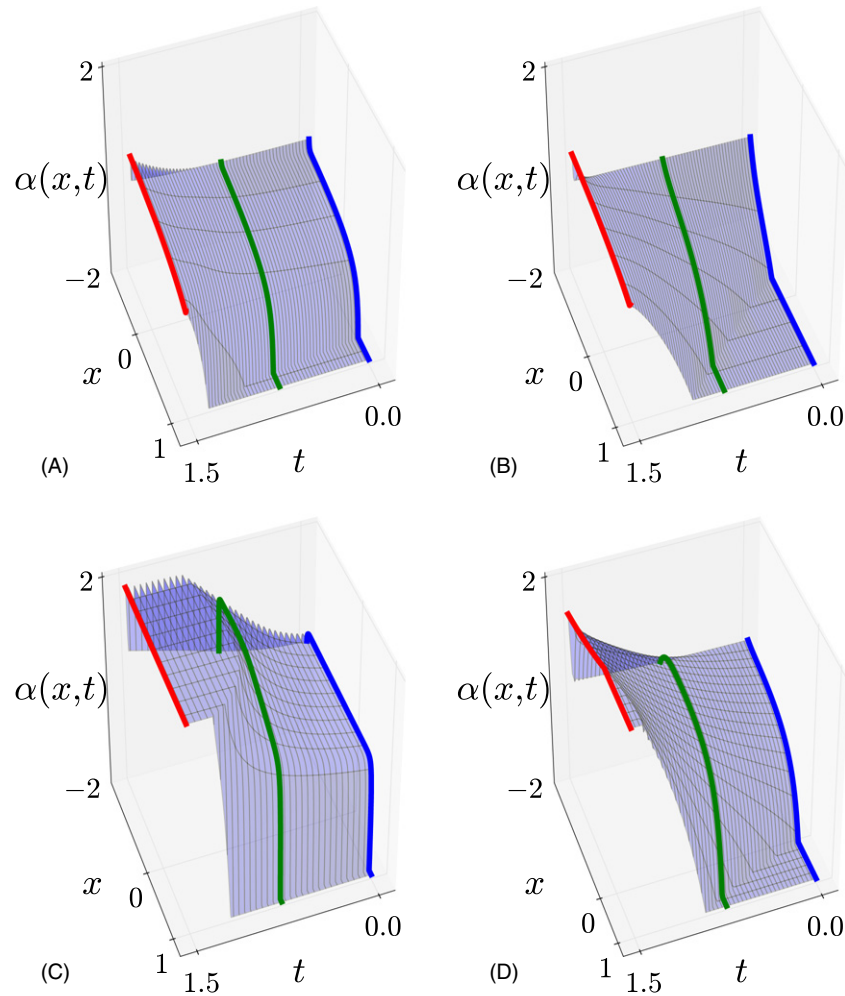
In both contexts, open-loop and closed-loop, finding the optimal control is computationally intensive as we numerically solve PDEs. If these algorithms were to be practical in an online setting, some more work would have to be done in order to ensure they can be computed very quickly (in the millisecond range) or that they can be pre-computed. In that regards, the dynamic programming approach is significantly more computationally efficient, at least in the current formulation.

Our work has stayed close to the paradigm of [1] in trying to make the neuron spike at a particular time. Similar to them, we have also incorporated into our objective the minimization

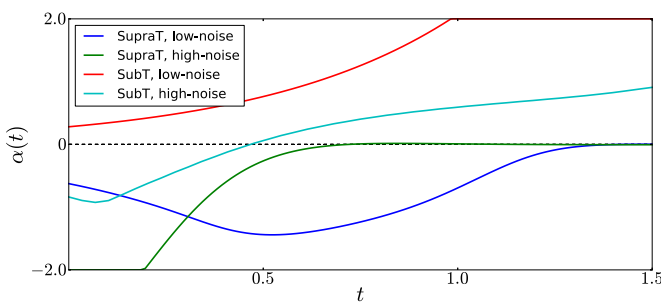
of total energy used to achieve our goal, which as discussed in [1] is sensible given the potential damage to the cell of accumulating charge. Furthermore, we have assumed that our control is constrained in magnitude, which is natural given physical limitations on equipment and safety considerations. Another possible constraint on the control is that it is charge-balanced, meaning that its time integral is zero,  $\int u dt = 0$ . Such constraints are most easily posed in the context of deterministic spiking models like the phase-response curve, see, e.g., Danzl *et al* [7]. It is possible to add them to the open-loop control, since the control is still deterministic, but it is non-trivial. It is even less trivial when one uses dynamic programming. In particular, insisting on charge-balance in our schemes will make it much more difficult to apply the TCs used in deriving both the closed- and open-loop controls.

Our computational scheme has made some assumptions about the physics of the controller. Most pertinently, we must be able to either measure the detailed voltage or at least the spikes of the controlled neuron and be able to react to them in real-time. This is needed both for system identification,





**Figure 14.** The effect of the energy parameter  $\epsilon$  on the closed-loop control. This is the same as figure 3, but for the higher value of  $\epsilon = 0.1$ . The desired spike time is set to  $t^* = 1.5$  and the control bounds are  $\alpha \in [-2, 2]$ . (A), (B) Supra-threshold-low-noise. (C), (D) Supra-threshold-high-noise. (E), (F) Sub-threshold-low-noise. (G), (H) Sub-threshold-high-noise.



**Figure 15.** The effect of the energy parameter  $\epsilon$  on the open-loop control. This is the same as figure 4, but with the higher value of  $\epsilon = 0.1$ . The desired spike time is set to  $t^* = 1.5$  and the bounds are  $\alpha \in [-2, 2]$ .

wherein we find the parameters for the LIF model, and for the actual control. We also need to be able to inject intra-cellular current into the cell. There exist alternatives to intra-cellular simulation, for example, [1] discusses how photo-stimulation can be used to elicit a similar effect, and our method is relevant to that context as well. Naturally, the ability to use

extra-cellular stimulation will be beneficial to the practical implementation of a neuron control strategy.

Our techniques as presented would suffer from the ‘curse of dimensionality’ if one were to try to control  $N$  neurons with known coupling, since indeed in this case, we would be solving  $N$ -dimensional PDEs for the control, be it closed- or open-loop. Such a scenario is further complicated by the requirements of system identification—one would have to identify the exact  $N$ -by- $N$  coupling strengths. However, in some situations, it may be possible to model network effects to first order by a rough approximation where the effects of other neurons (mean field effects) are taken into account by modifying the single-dimensional variables,  $\mu$ ,  $\beta$ —the bias current and the noise strength. As such, the control of  $N$  neurons could then be reduced coarsely to independently controlling each neuron, which is the technique described in this paper. We leave the exploration of this possibility for future work.

As points of outlook, we mention that our scheme can potentially be applied to produce a periodic train in the face of noise with applications to various biological pacemakers, or it can also be used to produce the opposite effect—random

behaviour at the single cell level, for example, by specifying a target train with Poisson-like characteristics.

The novelty in our work is primarily in obtaining a stochastic control scheme, which is not limited by the intensity of the noise in the system. In fact, by comparison with the naive ‘deterministic’ controller, we have shown how wrong the control strategy can be if we assume that the noise is negligible. Given this scope, the numerical demands on our scheme are non-trivial, but our experience and experiments show that these demands can be addressed. It however remains to be seen whether they can be successfully applied in the lab and beyond.

## Acknowledgments

SD is supported by the Danish Council for Independent Research | Natural Sciences. AL acknowledges support from NSERC Canada. AI is supported by a PGS-D scholarship from NSERC Canada. The work is part of the Dynamical Systems Interdisciplinary Network, University of Copenhagen.

## Appendix A. Calculating the terminal conditions for the HJB equation

Here, we explain how to calculate the first two moments of the remaining time-to-spike,  $\mathbb{E}[(T_{\text{sp}} - t^*)^n | X(t^*) = x, \alpha(t) = \alpha_{\text{max}}]$ ,  $n = 1, 2$ , which is needed for the TCs for  $w(x, t^*)$  and  $p(x, t^*)$  in, respectively, equations (12) and (15).

Let

$$\begin{aligned} T_{(1)}(x) &= \mathbb{E}[(T_{\text{sp}} - t^*) | X(t^*) = x, \alpha(t) = \alpha_{\text{max}}], \\ T_{(2)}(x) &= \mathbb{E}[(T_{\text{sp}} - t^*)^2 | X(t^*) = x, \alpha(t) = \alpha_{\text{max}}], \end{aligned} \quad (\text{A.1})$$

where as before,  $T_{\text{sp}} - t^*$  is the remaining time-to-spike in equation (1), with the voltage being  $X(t^*) = x$  at the desired spike time  $t = t^*$  and applying the control  $\alpha(t) = \alpha_{\text{max}}$  afterwards. Naturally, the  $T_{(i)}$  depend on  $x$  and are affected by the value of the various parameters,  $\alpha, \beta, \tau_c$ . We only need  $T_{(2)}$  for the TCs in equations (12) and (15), but  $T_{(1)}$  is needed as an auxiliary.

Analytical expressions exist for  $T_{(i)}$ , e.g. see [16]. However, in our experience, they are awkward to work with numerically, especially as they involve infinite alternating sums, which can easily overflow in numeric calculations. Moreover, they are not particularly convenient if you need  $T_{(i)}(x)$  for a wide range of  $x$  as we do here, since they calculate the value independently for each  $x$ . Instead, we will follow and slightly extend Jacobs<sup>4</sup> [17], in deriving and solving a differential equation for  $T_{(i)}(x)$ .

Basically, we are calculating the mean squared time to exit an interval  $[x_-, x_{th}]$ . Theoretically,  $x_- = -\infty$ , but for

<sup>4</sup> Jacobs only discusses how to calculate the first moment  $T_{(1)}$ ; however, it is quite clear from his discussion how to proceed recursively to obtain any higher moment  $T_{(i)}$ , by using previously calculated lower moments. This is what we do here. Moreover, the relation between mean exit times and boundary value problems is discussed in most introductory books on stochastic processes, see [27] for example.

our purposes, we will set it to some finite value and impose reflecting boundaries there just as we did for the value function  $w$ . This is a very good approximation for  $x_- \ll 0$ , as long as  $\alpha = \alpha_{\text{max}} > 0$ .

Let  $B(y, t|x) = \mathbb{P}[X_0 = y | X(t) = x]$  for a fixed  $x$ . Then  $B$  solves the following (backward Kolmogorov) PDE

$$-\partial_t B = \left( \mu + \alpha - \frac{y}{\tau_c} \right) \partial_y B - \frac{\beta^2}{2} \partial_y^2 B. \quad (\text{A.2})$$

Let  $\bar{G}(t, y)$  be the survival function for  $T_{\text{sp}}$  given  $X(t^*) = y$ ,

$$\begin{aligned} \bar{G}(t, y) &= \mathbb{P}[t^* > t | X(t^*) = y] \\ &= \mathbb{P}[X(s)|_{s \leq t} \in [x_-, x_{th}] | X(t^*) = y]. \end{aligned}$$

Because the dynamics do not depend explicitly on time, we have

$$\bar{G}(t, y) = \int_{x_-}^{x_{th}} B(y, t|x) dx$$

and so  $\bar{G}$ , being a definite integral of  $B$ , satisfies the same PDE:

$$-\partial_t \bar{G} = \left( \mu + \alpha - \frac{y}{\tau_c} \right) \partial_y \bar{G} - \frac{\beta^2}{2} \partial_y^2 \bar{G}, \quad (\text{A.3})$$

with the BCs, ICs:

$$\begin{cases} \bar{G}(t, x_{th}) = 0 & (T_{\text{sp}} = t \Rightarrow \mathbb{P}[T_{\text{sp}} > t] = 0) \\ \partial_y \bar{G}(t, x_-) = 0 & (\text{lower BC}) \\ \bar{G}(t^*, y) = 1. & (T_{\text{sp}} \text{ is definitely } > \tau \text{ for } X(t^*) \in (x_-, x_{th})). \end{cases} \quad (\text{A.4})$$

If we solve the PDE for  $\bar{G}$  over  $t \in [0, \infty)$ , then we can calculate  $T_{(2)}(x)$  as

$$T_{(2)}(x) = \int_0^\infty t^2 \cdot (-\partial_t \bar{G}(t, x)) dt \quad (\text{A.5})$$

$$= \int_0^\infty 2t \cdot \bar{G}(t, x) dt, \quad (\text{A.6})$$

assuming it exists.

Since  $T_{(2)}$  is an integral over time, we can reduce the PDE in equation (A.3) for the distribution to an ordinary differential equation (ODE) for the moments by integrating over time. Multiply both sides of equation (A.3) by  $2t$ , and integrate with respect to time:

$$\begin{aligned} &\int_0^\infty 2t (-\partial_t \bar{G}) dt \\ &= \int_0^\infty 2t \left[ \left( \mu + \alpha_{\text{max}} - \frac{y}{\tau_c} \right) \partial_y \bar{G} - \frac{\beta^2}{2} \partial_y^2 \bar{G} \right] dt. \end{aligned} \quad (\text{A.7})$$

The left-hand side turns out to be  $2T_{(1)}$ , so

$$2T_{(1)} = \left( \mu + \alpha_{\max} - \frac{y}{\tau_c} \right) \partial_y T_{(2)} - \frac{\beta^2}{2} \partial_y^2 T_{(2)}. \quad (\text{A.8})$$

Similarly, we can derive an equation for  $T_{(1)}$ , namely:

$$1 = \left( \mu + \alpha_{\max} - \frac{y}{\tau_c} \right) \partial_y T_{(1)} - \frac{\beta^2}{2} \partial_y^2 T_{(1)}. \quad (\text{A.9})$$

In both cases, the boundary conditions will be

$$\{T_{(i)}(x_{th}) = 0 \quad (\text{we are already spiking}). \quad (\text{A.10})$$

Equations (A.9) and (A.8) can be reduced to first order differential equations and solved analytically, at least up to some definite integral, but the analytical solution is not particularly useful or necessary for numerical calculations. Instead, we opt to calculate the solutions to (A.9) and (A.8) directly via an ODE solver, starting from  $T_i(x_{th}) = 0$  and integrating down for  $x \in [x_-, x_{th}]$ . That is how we have obtained the TCs for both  $w(x, t^*)$  and  $p(x, t^*)$  to apply in, respectively, equations (12) and (15).

## Appendix B. Gradient descent algorithm for the maximum principle

---

### Algorithm B1. Gradient descent algorithm for obtaining the optimal open-loop control

---

```

Fix  $t^*, \epsilon, \dots$  the problem parameters
Fix  $\{t_n\}_0^{N_t}$  a time-discretization of  $[0, t^*]$ 
Fix  $g_{\text{tol}}$  a convergence tolerance for the gradient
Fix  $K_{\text{max}}, J_{\text{max}}$  number of maximum iterations in outer, resp. inner descent loops
Fix  $s$  the initial step-size.
# we use  $g_{\text{tol}} = 1e-5, K_{\text{max}} = 100, J_{\text{max}} = 10, s = 10$ 
 $\alpha_1(t) \leftarrow (\alpha_{\max} - \alpha_{\min}) \cdot t/t^* + \alpha_{\min}$ 
 $\# \alpha_1(t) \sim$  initial guess for the control, linear interpolate between  $\alpha_{\min}, \alpha_{\max}$ 
for  $k = 1 \dots K_{\text{max}}$  do
  # This is the outer loop where we descend down different gradients
  Calculate  $f_k, J_k, p_k, \delta_\alpha J_k$  corresponding to  $\alpha_k$  from equations (13)–(15) and (17)
   $N_{\text{active}} \leftarrow$  Number of time nodes  $t_n$ , where either  $\alpha_k(t_n) \neq \{\alpha_{\min}, \alpha_{\max}\}$  or  $\delta_\alpha J_k(t_n)$ 
  points inwards
  if  $\|\delta_\alpha J_k\|_{\mathbb{R}^{N_{\text{active}}}} \leq g_{\text{tol}} \cdot N_{\text{active}}$  then
    # 'Active' gradient is small enough, consider converged:
    BREAK
  end if
  # Find the step size,  $s$ , of how far to move  $\alpha$  in the direction  $\delta_\alpha J_k$ :
  for  $j = 1 \dots J_{\text{max}}$  do
    # This is the inner loop where we find how much to descend down the current gradient
     $\alpha_{k,j} \leftarrow \alpha_k - s_j \cdot \delta_\alpha J_k$ 
    #  $\alpha_{k,j}$  is the new control strategy to try
    Calculate  $f_{k,j}, J_{k,j}$  corresponding to  $\alpha_{k,j}$  from equations (13) and (14)
    # Recall  $f_{k,j}$  is a probability density resulting from the control  $\alpha_{k,j}$  and  $J_{k,j}$  is the objective value resulting from  $f_{k,j}$ 
    if  $J_{k,j} < J_k$  then
      # We found a better (smaller) objective value
       $s \leftarrow 2s_j$  # try a more aggressive step in the next iteration
      BREAK
    end if
    if  $j == J_{\text{max}}$  then
      # We exhausted the step search without finding a smaller  $J$ , return current values
      Return  $J_k, \alpha_k$ 
    end if
  # Continue the inner loop, now try with a smaller step:
   $s_{j+1} \leftarrow s_j/2$ .
end for # single step loop
if  $k == K_{\text{max}}$  then
  ERROR 'Could not converge'
end if
# Assign the new candidate for the optimal control and re-loop
 $\alpha_{k+1} \leftarrow \alpha_{k,j}$ 
end for # gradient descent loop
return  $J_k, \alpha_k$ 

```

---

## References

- [1] Ahmadian Y, Packer A M, Yuste R and Paninski L 2011 Designing optimal stimuli to control neuronal spike timing *J. Neurophysiol.* **106** 1038–53
- [2] Annunziato M and Borzi A 2010 Optimal control of probability density functions of stochastic processes *Math. Modelling Anal.* **15** 393–407
- [3] Annunziato M and Borzi A 2013 A Fokker–Planck control framework for multidimensional stochastic processes *J. Comput. Appl. Math.* **237** 487–507
- [4] Berg R W and Ditlevsen S 2013 Synaptic inhibition and excitation estimated via the time constant of membrane potential fluctuations *J. Neurophysiol.* **110** 1021–34
- [5] Borzi A and Schulz V 2012 *Computational Optimization of Systems Governed by Partial Differential Equations* (Philadelphia, PA: SIAM)
- [6] Danzl P, Hespanha J and Moehlis J 2009 Event-based minimum-time control of oscillatory neuron models: phase randomization, maximal spike rate increase, and desynchronization *Biol. Cybern.* **101** 387–99
- [7] Danzl P, Nabi A and Moehlis J 2010 Charge-balanced spike timing control for phase models of spiking neurons *Discrete Continuous Dyn. Syst.* **28** 1413–35
- [8] Dasanayake I and Jr-Shin L 2011 Constrained minimum-power control of spiking neuron oscillators *IEEE Conf. on Decision and Control and European Control Conf.* pp 3694–9
- [9] Ditlevsen S and Greenwood P 2013 The Morris–Lecar neuron model embeds a leaky integrate-and-fire model *J. Math. Biol.* **67** 239–59
- [10] Ditlevsen S and Lánský P 2007 Parameters of stochastic diffusion processes estimated from observations of first-hitting times: application to the leaky integrate-and-fire neuronal model *Phys. Rev. E* **76** 41906
- [11] Ditlevsen S and Samson A 2014 Estimation in the partially observed stochastic Morris–Lecar neuronal model with particle filter and stochastic approximation methods arXiv:1207.1865
- [12] Feng J and Tuckwell H 2003 Optimal control of neuronal activity *Phys. Rev. Lett.* **91** 1–4
- [13] Feng X-J, Greenwald B, Rabitz H, Shea-Brown E and Kosut R 2007 Toward closed-loop optimization of deep brain stimulation for Parkinson’s disease: concepts and lessons from a computational model *J. Neural Eng.* **4** L14–21
- [14] Fleming W H and Rishel R W 1975 *Deterministic and Stochastic Optimal Control* (Berlin: Springer)
- [15] Gerstner W and Kistler W M 2002 *Spiking Neuron Models: Single Neurons, Populations, Plasticity* (Cambridge: Cambridge University Press)
- [16] Inoue J, Sato S and Ricciardi L M 1995 On the parameter estimation for diffusion models of single neuron’s activities *Biol. Cybern.* **73** 209–21
- [17] Jacobs K 2010 *Stochastic Processes for Physicists: Understanding Noisy Systems* (Cambridge: Cambridge University Press)
- [18] Jones E et al 2001 SciPy: Open source scientific tools for Python [www.scipy.org](http://www.scipy.org)
- [19] Lee S 1994 Optimal drift on  $[0, 1]$  *Trans. Am. Math. Soc.* **346** 159–75
- [20] Lefebvre M 1987 Optimal control of an Ornstein–Uhlenbeck process *Stoch. Process. Appl.* **24** 89–97
- [21] Moehlis J, Shea-Brown E and Rabitz H 2006 Optimal inputs for phase models of spiking neurons *J. Comput. Nonlinear Dyn.* **1** 358–67
- [22] Morris C and Lecar H 1981 Voltage oscillations in the barnacle giant muscle fiber *Biophys. J.* **35** 193–213
- [23] Nabi A, Mirzadeh M, Gibou F and Moehlis J 2013 Minimum energy desynchronizing control for coupled neurons *J. Comput. Neurosci.* **34** 259–71
- [24] Nabi A and Moehlis J 2011 Single input optimal control for globally coupled neuron networks *J. Neural Eng.* **8** 065008
- [25] Nabi A and Moehlis J 2012 Time optimal control of spiking neurons *J. Math. Biol.* **64** 981–1004
- [26] Nabi A, Stigen T, Moehlis J and Netoff T 2013 Minimum energy control for *in vitro* neurons *J. Neural Eng.* **10** 036005
- [27] Oksendal B K 2007 *Stochastic Differential Equations: An Introduction with Applications* 6th edn (Berlin: Springer)
- [28] Press W H, Flannery B P, Teukolsky S A and Vetterling W T 1992 *Numerical Recipes in C: The Art of Scientific Computing* 2nd edn (Cambridge: Cambridge University Press)
- [29] Rinzel J and Ermentrout G B 1989 Analysis of neural excitability and oscillations *Methods in Neural Modeling* (Cambridge, MA: MIT Press)
- [30] Tatenko T and Pakdaman K 2004 Random dynamics of the Morris–Lecar neural model *Chaos* **14** 511–30
- [31] Whittle P 1996 *Optimal Control: Basics and Beyond* (New York: Wiley)
- [32] Yongjun W, Peng J and Luo M 2009 First spiking dynamics of stochastic neuronal model with optimal control *Advances in Neuro-Information Processing (Lecture Notes in Computer Science vol 5506)* ed M Köppen, N Kasabov and G Coghill (Berlin: Springer) pp 129–36